



**SISTEMA DE INFORMACIÓN GEOGRÁFICA EN LA WEB (WEBGIS) PARA EL
APOYO A LA VIGILANCIA ENTOMOLÓGICA DE AEDES AEGYPTI**

JORGE LUIS GOYENECHE MORA

**UNIVERSIDAD DE LOS LLANOS
FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍA
ESCUELA DE INGENIERIA
PROGRAMA DE INGENIERÍA DE SISTEMAS
VILLAVICENCIO, COLOMBIA**

2018

**SISTEMA DE INFORMACIÓN GEOGRÁFICA EN LA WEB (WEBGIS) PARA EL
APOYO A LA VIGILANCIA ENTOMOLÓGICA DE AEDES AEGYPTI**

JORGE LUIS GOYENECHE MORA

**Trabajo de grado en modalidad EPI presentado como requisito parcial para
optar al título de Ingeniero de Sistemas**

Director:

Ph. D(c). M.Sc., Ing., Cesar Augusto Diaz Celis

Codirector:

Ph. D. M.Sc., Ing., Ángel Alfonso Cruz Roa

**UNIVERSIDAD DE LOS LLANOS
FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍA
ESCUELA DE INGENIERIA
PROGRAMA DE INGENIERÍA DE SISTEMAS
VILLAVICENCIO, COLOMBIA
2018**

NOTA DE ACEPTACIÓN

ING. CESAR AUGUSTO DIAZ CELIS, M.Sc,
Ph.D(c)
Jurado

“Dedicado a mis padres, que con su esfuerzo me lo han dado todo en la vida y me han apoyado siempre. A mi hermana Lina por siempre acompañarme y brindarme su cariño”.

Jorge Luis Goyeneche Mora

AGRADECIMIENTOS

Agradecimientos a mi director y codirector de proyecto, el Ing. Cesar Augusto Diaz Celis, M. Sc, Ph. D(c) y el Ing. Ángel Alfonso Cruz Roa por compartirme su conocimiento, orientarme a lo largo de todo el proyecto de grado y ser pacientes conmigo. Mi total admiración y respeto.

Gracias también al Profesional en Enfermería Oscar Alexander Gutiérrez Lesmes, M. Sc, cuyos conocimientos fueron de gran ayuda en el desarrollo de este proyecto de grado.

Al Ing. Daniel Fernando Muñoz Meléndez, por su buena disposición y entrega a la hora de trabajar juntos. También, a los compañeros Diana y Ricardo por su ayuda en el proyecto.

Muchas gracias a los compañeros del grupo de investigación GITECX y del grupo de estudio AdaLab, quienes muchas veces escucharon mis exposiciones acerca del proyecto con el fin de ayudarme en este proceso.

A cada profesor de quien recibí clases hasta este punto de mí vida. Sin su ayuda en este largo proceso habría sido imposible llegar hasta aquí.

Para terminar, a mi familia, amigos, compañeros y a todas aquellas personas que me han acompañado en este proceso.

¡Gracias!

TABLA DE CONTENIDO

AGRADECIMIENTOS	5
TABLA DE CONTENIDO	6
INTRODUCCIÓN	9
OBJETIVOS.....	10
Objetivo General:.....	10
Objetivos Específicos:.....	10
MARCO DE REFERENCIA DEL PROYECTO:	11
Enfermedades de Transmisión Vectorial (ETV):	11
Aedes Aegypti:.....	12
Aplicaciones Web:	12
• Capa del navegador:.....	13
• Capa del servidor:	13
• Capa de persistencia:	13
Sistemas de información geográfica (SIG):.....	14
WebGIS:	14
ANTECEDENTES.....	15
METODOLOGIA	17
1. Levantamiento de requerimientos.	17
2. Diseño de la solución.	18
Herramientas tecnológicas:	18
Funcionalidades del sitio web:.....	22
Funcionalidades de la API:	25
Diseño del modelo Entidad-Relación	28
3. Implementación de la solución.	33

4. Pruebas.	42
5. Corrección de errores.	42
6. Realización de prueba piloto.	45
RESULTADOS.....	50
CONCLUSIONES	51
TRABAJOS FUTUROS.....	52
REFERENCIAS BIBLIOGRÁFICAS.	53

RESUMEN

El vector *Aedes Aegypti* es el principal transmisor de enfermedades como el Dengue, el Zika y el Chikunguña en Colombia. Debido a que estas enfermedades representan una importante carga en el sistema de salud y la economía de dicho país, se evidencio que era necesaria la elaboración de herramientas que ayudaran a combatir dicho vector. El sistema de información geográfica en la Web (WebGIS) para el apoyo a la vigilancia entomológica de *Aedes Aegypti* (llamado VIGEA) se desarrolló con el fin de ofrecer una herramienta y suplir dicha necesidad. Para el desarrollo del proyecto, se hizo uso de herramientas web como Tomcat, Nginx y Geoserver; para el caso del desarrollo se usó Python como lenguaje de programación y Django como marco de trabajo Modelo-Vista-Controlador. Como resultado, se obtuvo una aplicación web que permite crear mapas de infestación a partir de información compartida por la comunidad de manera colaborativa, dicha información, es validada por expertos con el objetivo de solo contar con información veraz al momento en que la aplicación web construye los mapas de infestación. Para concluir, se evidencio que los mapas de infestación contruidos por la aplicación Web permiten evidenciar fácilmente lugares geográficos en los cuales exista potencialmente el vector y en los que los entes encargados podrían tomar acciones.

Palabras clave: *Aedes Aegypti*, Vigilancia Entomológica, CrowdSourcing, WebGIS

INTRODUCCIÓN

El vector *Aedes Aegypti* es una especie antropofílica que se ha adaptado fácilmente a entornos urbanos, debido a que, en estos entornos, las hembras de dicha especie tienen la facilidad de picar múltiples veces y encontrar fácilmente recipientes artificiales en los que depositar sus huevos, adicionalmente, estos huevos tienden a ser bastante resistentes a la desecación, lo que aumenta las posibilidades de la reproducción[1]. Este mosquito, es el principal transmisor de enfermedades como el Dengue, el Zika y el Chikunguña, lo que representa una importante carga al sistema de salud y a la economía[2].

La vigilancia en salud pública es la gestión (es decir, recolección, análisis e interpretación) de datos de gran importancia para prevenir enfermedades[3]. Para poder ejercer esta vigilancia, son necesarios sistemas de vigilancia en los cuales apoyarse a la hora de desarrollar dicha tarea. Un sistema de vigilancia puede recoger los síntomas y/o las características de una enfermedad o un suceso de salud pública casi en tiempo real, esto, ayudaría a detectar rápidamente una etapa temprana de un brote[3]. Debido a los beneficios que tienen los sistemas de vigilancia en salud pública, pueden ser usados como herramienta de apoyo, proporcionando a los entes encargados la capacidad de establecer políticas de prevención a largo plazo para poder responder a posibles brotes de manera rápida[4].

Evidenciando la necesidad de herramientas que apoyen la toma de decisiones en el control del mosquito *Aedes Aegypti*, se realizó una revisión de literatura con el fin de encontrar proyectos o artículos relacionados al tema de interés. A partir de esta evidencia científica se consideró que las herramientas tecnológicas en general son de gran ayuda a la hora de proporcionar apoyo en la toma de decisiones en temas relacionados a la salud pública. También, permitió identificar que los GIS y los WebGIS son herramientas de gran ayuda en el seguimiento a distintos tipos de enfermedades, ofreciendo grandes beneficios, como la posibilidad de gestionar mejor la información de una enfermedad o visualizar de una manera más práctica información relevante para la toma de decisiones.

Debido a la carencia de dichas herramientas en la región enfocadas a la prevención y vigilancia entomológica del mosquito *Aedes Aegypti*, se estableció un plan de trabajo con el objetivo de diseñar e implementar un sistema de información geográfica en la Web (WebGIS) para la recolección, representación y visualización de datos georreferenciados de criaderos de larvas y/o pupas de *Aedes Aegypti* como apoyo a la vigilancia entomológica y de esta manera obtener mapas de infestación en una unidad territorial(mapas de calor).

OBJETIVOS

Objetivo General:

- Desarrollar un sistema de información geográfica en la Web (WebGIS) con información relacionada de criaderos con larvas y/o pupas de Aedes Aegypti como apoyo a la vigilancia entomológica.

Objetivos Específicos:

- Diseñar e implementar un sistema de información geográfica en la Web (WebGIS) para la recolección, representación y visualización de los datos tomados en campo por medio de una aplicación móvil.
- Integrar el sistema de información geográfica (Web GIS) con la aplicación móvil (mHealth).
- Generar un mapa de infestación del Aedes Aegypti en una unidad territorial con base en los datos almacenados.

MARCO DE REFERENCIA DEL PROYECTO:

Enfermedades de Transmisión Vectorial (ETV):

“Las enfermedades causadas por patógenos y parásitos en poblaciones humanas propagadas por artrópodos a humanos son denominadas enfermedades de transmisión vectorial”[2]; dichas enfermedades ocurren debido a que los vectores y los seres humanos tienden a convivir bajo ciertos factores (humedad, temperatura, hacinamiento, altitud, etc.) en un mismo hábitat y como consecuencia terminan acarreando un importante número de enfermos a los sistemas de salud, al punto de llegar a representar una sexta parte de los casos de enfermedad generados en el mundo[1][2]. En la Figura 1 se muestra un ejemplo del proceso de expansión del virus del dengue mediante el vector *Aedes Aegypti*.

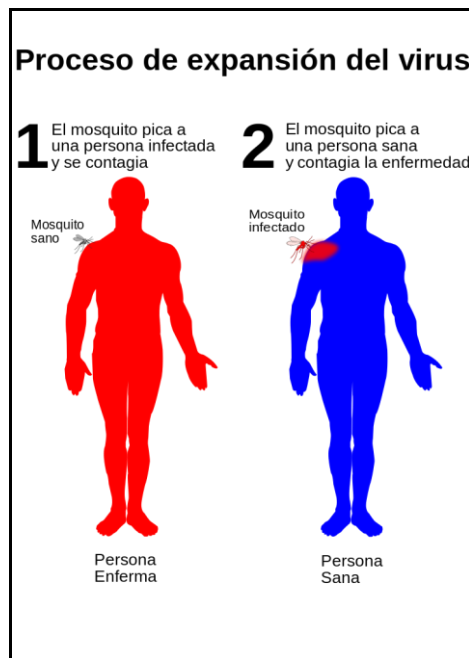


Figura 1. Proceso de expansión del virus del dengue

Tomado de:

https://upload.wikimedia.org/wikipedia/commons/thumb/7/7a/Contagio_Dengue.svg/744px-Contagio_Dengue.svg.png

Aedes Aegypti:

El vector *Aedes Aegypti* (ver Figura 2) es una especie antropofílica que se ha adaptado fácilmente a entornos urbanos, debido a que, las hembras de dicha especie tienen la facilidad de picar múltiples veces y encontrar fácilmente recipientes artificiales en los que depositar sus huevos, los cuales tienden a ser bastante resistentes a la desecación, lo que aumenta las posibilidades de la reproducción del mosquito[1].



Figura 2. *Aedes Aegypti*

Tomado de:

https://upload.wikimedia.org/wikipedia/commons/thumb/d/d0/Aedes_aegypti.jpg/1024px-Aedes_aegypti.jpg

Aplicaciones Web:

Una aplicación Web es una herramienta de software a la que se accede mediante el uso de Internet. “Es un tipo de aplicación, cliente/servidor, donde tanto el cliente (navegador, explorador o visualizador) como el servidor (servidor Web) y el protocolo mediante el que se comunican (*HyperText Transfer Protocol* o *HTTP*) están estandarizados” [5] (ver Figura 3).

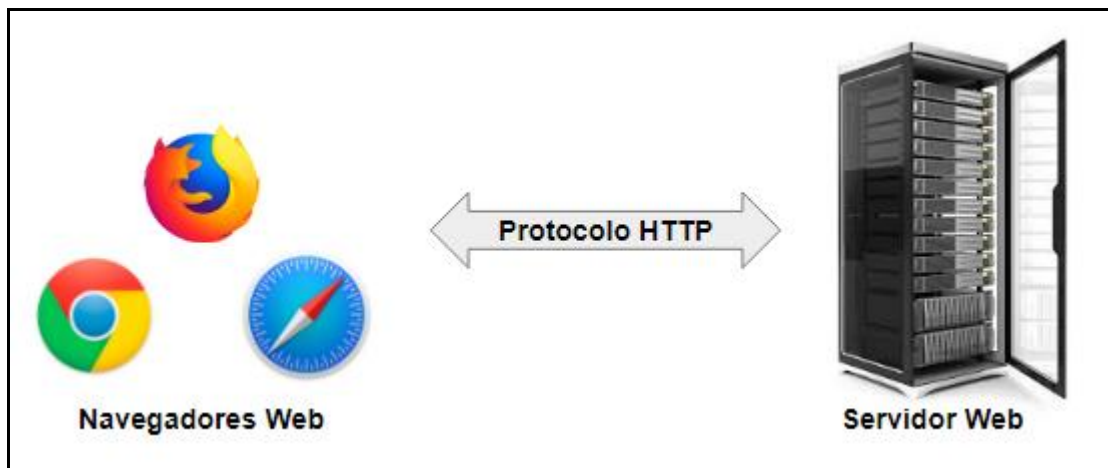


Figura 3. Aplicación Web

En el uso de aplicaciones Web intervienen dos partes (el cliente y el servidor Web) y un protocolo de comunicación (HTTP), se considera habitualmente, que dichas aplicaciones tienen una estructura compuesta por tres capas.

- **Capa del navegador:** Es la encargada de interactuar con los usuarios. Está conformada por software o scripts que pueden ser interpretados por los navegadores Web (Por ejemplo: HTML, JavaScript) con el fin de facilitar la manera en que se comunica el usuario y el servidor Web[5].
- **Capa del servidor:** Es el software encargado de procesar todas las solicitudes que hace el cliente. A pesar de que actualmente, muchas de las cosas que se realizan en una aplicación Web pueden ser procesadas por la capa del navegador, generalmente, es la capa del servidor quien ejecuta la mayor parte de las tareas que son ejecutadas por el usuario. Los lenguajes más comunes en la capa del servidor son: PHP, Python, Java y C#.
- **Capa de persistencia:** Lugar donde se almacenan los datos generados por la aplicación Web, generalmente, una base de datos.

En la Figura 4, se observa la forma en que interactúan las capas que conforman la estructura de una aplicación Web.



Figura 4. Estructura de una aplicación Web

Sistemas de información geográfica (SIG):

Los sistemas de información geográfica (SIG) son usados para la recolección, almacenamiento, procesamiento y visualización de datos que están espacialmente referenciados. Durante los últimos años, debido a que la capacidad de procesamiento de cómputo ha aumentado de manera significativa, se han empezado a explorar las capacidades que tienen los SIG en múltiples áreas de estudio, una de ellas es la salud, algunos autores sugieren que los SIG podrían ser una importante herramienta en el apoyo a la toma de decisiones y en la vigilancia de enfermedades, debido a que, estos tienen la capacidad de almacenar grandes cantidades de información heterogénea espacialmente referenciada de múltiples fuentes, lo que permitiría a expertos tener fácil acceso a múltiple información a partir de un único sistema, sin embargo, también se han encontrado falencias, una de ellas es la fragilidad que tienen los SIG a la hora de procesar información no veraz, como consecuencia, se podría considerar que la utilidad de un SIG es directamente proporcional a la veracidad de la información que este contiene[2][6][7][8][9][10][11][12].

WebGIS:

Con el rápido desarrollo de la Web en los últimos años, sumado a las bondades de los SIG, ha empezado a volverse popular el término WebGIS (Sistemas de información geográfica basados en la Web), los cuales divisan un importante potencial ocasionado por su capacidad para complementarse entre sí, ya que una de las ventajas de los SIG es su capacidad para procesar información y con la ayuda de la Web, estos podrían fácilmente compartir dicha información procesada[10].

ANTECEDENTES

Evidenciando la necesidad de herramientas que apoyen la toma de decisiones en el control del mosquito *Aedes Aegypti*, se realizó una revisión de literatura con el fin de encontrar trabajos previos, proyectos o artículos científicos relacionados al tema de interés, como resultado se hallaron los siguientes antecedentes:

En [13], se describe el funcionamiento de un sistema de vigilancia de enfermedades transmisibles en Suecia, el cual, tiene como objetivo el almacenamiento de información relacionada a estas enfermedades con el fin de servir de herramienta de apoyo en la toma de decisiones de los entes encargados de la salud en dicho país.

En [14], el autor relata la manera en que se desarrolló un sistema de información geográfica en la India, con el fin de monitorear las poblaciones del *Aedes Aegypti* y funcionar como herramienta de apoyo en el control de dicha especie.

En [9], se relata la manera en que se creó un sistema de información geográfica basado en la Web (WebGIS) que se desarrolló en Japón, con el fin de obtener información geolocalizada de enfermedades infecciosas. Los autores explican la necesidad de contar con una herramienta que le permita gestionar, analizar y visualizar información relacionada a dichas enfermedades para ofrecer apoyo a los médicos y profesionales de la salud.

En [10], los autores relatan un caso de éxito de un sistema de información geográfica creado en New Brunswick (Canadá) y Maine (EE. UU.) para cartografiar información relacionada a enfermedades. A través de este caso de estudio los autores destacan la efectividad de esta herramienta debido a que permite el intercambio de datos y el análisis de estos, y también, ofrecer mejoras en la eficiencia y eficacia de la vigilancia de la salud pública.

En [15], los autores describen la construcción de un sistema de información geográfica basado en la Web (WebGIS) en Francia, con el fin de ofrecer acceso a la epidemiología de la demanda y la oferta de atención con relación a enfermedades renales. Su fin es servir como herramienta de visualización y apoyo a la toma de decisiones.

En [11], los autores relatan la manera en que se hace uso de un sistema de información geográfica con el fin de cartografiar la densidad del vector *Aedes Aegypti* en una determinada unidad territorial en India. Destacan que mediante el uso de esta herramienta se obtuvo información relevante acerca de las variables socioeconómicas y ambientales que están asociadas a las epidemias de Dengue y Chikunguña en la India.

En [11], este artículo destaca la importancia de las herramientas tecnológicas y del papel que han jugado estas en el cuidado de la salud pública. Concluye que desde sus inicios las herramientas tecnológicas han sido de gran utilidad debido a que han permitido descubrir y contener la propagación de enfermedades de manera oportuna, como consecuencia han ayudado a la preservación de miles de vidas y a reducir los impactos financieros.

METODOLOGIA

Este proyecto constó de una metodología conformada por ocho etapas para el desarrollo del mismo, las cuales se presentan a continuación:

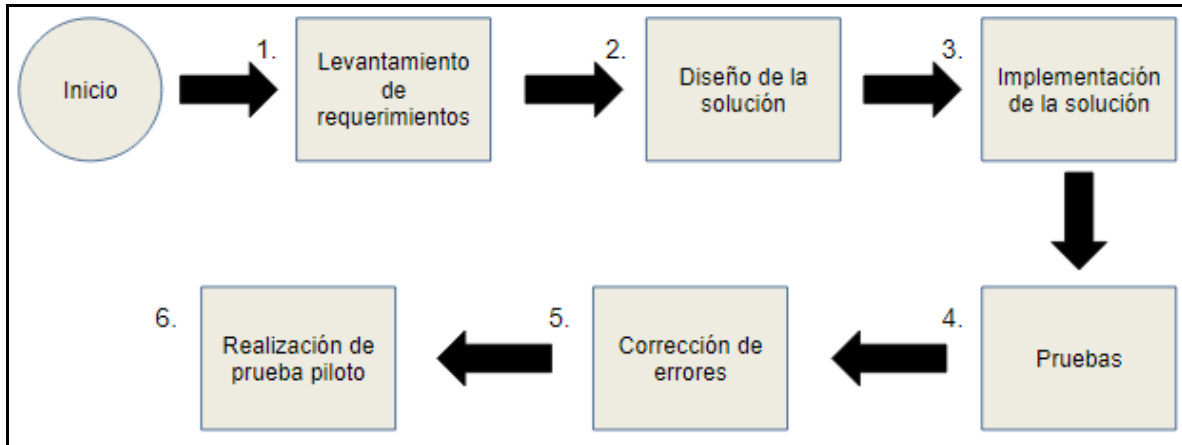


Figura 5. Etapas de la metodología de desarrollo del proyecto de grado.

1. Levantamiento de requerimientos.

Para la realización de este proyecto, como medida inicial, se realizó el levantamiento de requerimientos necesarios para la construcción del WEBGIS. Dicho levantamiento de requerimientos se realizó mediante la orientación de los Ingenieros Ángel Cruz y Cesar Díaz, y el Profesional en Enfermería Oscar Gutiérrez. A partir de las necesidades se decidió establecer los siguientes requerimientos:

- El sistema debe permitir la gestión de usuarios (Creación, Edición, Lectura y Eliminación).
- El sistema debe permitir la gestión de reportes: Para la gestión de reportes se identificó la necesidad de establecer tres diferentes tipos de reporte: criaderos, estadios y síntomas. Para cada uno de estos reportes se debía poder hacer la respectiva gestión (Creación, Edición, Lectura y Eliminación).
- El sistema debe permitir la creación de mapas de infestación visibles a través de la web: para la creación de dichos mapas se establecieron las siguientes necesidades:
 - Crear mapa de infestación a partir de los reportes de criaderos.

- Crear mapa de infestación a partir de cada uno de los tipos de criadero.
- Crear mapa de infestación a partir de los reportes de estadios.
- Crear mapa de infestación a partir de cada uno de los estadios.
- Crear mapa de infestación a partir de los reportes de síntomas.
- Crear mapa de infestación a partir de cada uno de los síntomas existentes en los reportes de síntomas.
- El sistema debe permitir la creación de roles (Administrador, gestor de contenidos, expertos) para segmentar el acceso a contenido del sistema.
- El sistema debe permitir que expertos en entomología puedan tener acceso a la información de los reportes, para que, a partir de su conocimiento, establezcan cuáles reportes son válidos y cuáles no.
- El sistema debe contar con una API que le permita comunicarse con una aplicación móvil. Para la API se generaron los siguientes requerimientos:
 - Autenticación.
 - Lectura de datos de un usuario.
 - Crear reportes de estadio
 - Crear reportes de criadero
 - Crear reportes de síntoma
 - Lectura de reportes de un usuario
 - Recuperación de contraseña de un usuario
 - Cambio de contraseña de un usuario

2. Diseño de la solución.

A partir de las necesidades establecidas por un experto y con ayuda del director y codirector del proyecto, se inició una investigación de herramientas tecnológicas en las cuales pudiéramos apoyarnos para alcanzar los objetivos de este proyecto. Por lo que, finalizada la investigación se decidió hacer uso de las siguientes herramientas tecnológicas:

Herramientas tecnológicas:

a. PostgreSQL:

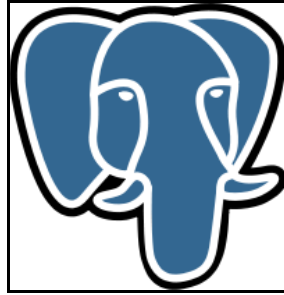


Figura 6. Logotipo PostgreSQL.

Tomado de: <https://es.wikipedia.org/wiki/PostgreSQL>

“Es un potente sistema de base de datos relacional de objetos abierto que utiliza y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de forma segura las cargas de trabajo de datos más complicadas”[16]. Este sistema gestor de bases de datos fue seleccionado debido a que tiene una buena reputación respecto a su confiabilidad e integridad de los datos. Cuenta con una extensión que le permite el almacenamiento de objetos geográficos y, además, es impartido en la materia bases de datos del programa de Ingeniería de Sistemas de la Universidad de los Llanos.

b. PostGIS:

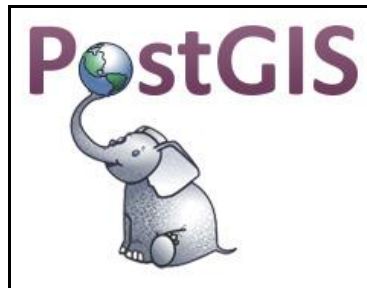


Figura 7. Logotipo PostGIS.

Tomado de: <https://es.wikipedia.org/wiki/PostGIS>

“Es una extensión de bases de datos espacial para la base de datos relacional de objetos PostgreSQL. Agrega soporte para objetos geográficos que permiten que las consultas de ubicación se ejecuten en SQL”[16].

c. Tomcat:

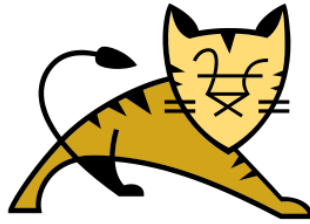


Figura 8. Logotipo Apache Tomcat.

Tomado de: <https://es.wikipedia.org/wiki/Tomcat>

Es un servidor web multiplataforma que sirve como contenedor de servlets[17], esta herramienta se usó en el proyecto con el fin de alojar GeoServer.

d. GeoServer:



Figura 9. Logotipo Geoserver.

Tomado de: <https://boundlessgeo.com/2017/03/geoserver-gis-geoserver-code-sprint-needs-support/>

“Es un servidor de software basado en Java que permite a los usuarios ver y editar datos geoespaciales. Mediante el uso de estándares abiertos establecidos por el Open Geospatial Consortium (OGC), GeoServer permite una gran flexibilidad en la creación de mapas y el intercambio de datos”[18]. GeoServer fue usado debido a que permitía representar de manera fácil datos geográficos y en distintos formatos de salida.

e. OpenLayers:

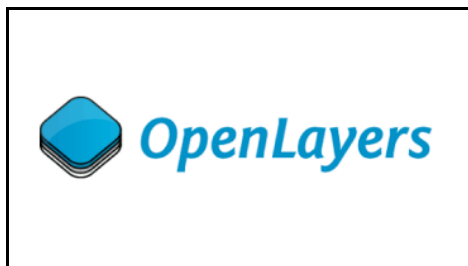


Figura 10. Logotipo OpenLayers.

Tomado de: <https://www.blogsig.com/2018/04/que-es-openlayers-y-como-usar-el-codigo.html>

Es una librería de JavaScript que permite mapeo en la web mediante el consumo de datos en distintos formatos[19], por lo que con la ayuda de GeoServer son de gran importancia en este proyecto debido a que mediante este, se le permite al usuario observar los mapas de calor a partir de los mapas de calor.

f. Nginx



Figura 11. Logotipo Nginx.

Tomado de: <https://es.wikipedia.org/wiki/Nginx>

Es un servidor web[20]. Fue usado con el fin de servir a través de este el software desarrollado en Python mediante el marco de trabajo Django.

g. Django:



Figura 12. Logotipo Django.

Tomado de: <https://www.djangoproject.com/s/img/logos/django-logo-positive.png>

Es un marco de trabajo de desarrollo web escrito en Python, de código abierto, basado en un patrón de diseño denominado Modelo-Vista-Plantilla. Fue seleccionado como framework de desarrollo web para este proyecto debido a que por defecto este cuenta con una API que permite al desarrollador manejar bases de datos geoespaciales.

A partir de la elección de las herramientas de desarrollo se empezaron a definir un conjunto de tareas que debía tener el software resultante. Como punto de inicio, se fracciono el software en dos secciones:

1. El sitio web en donde los usuarios podrían hacer gestión de los datos o visualizar los mapas de calor.
2. La API, que sería la encargada de establecer comunicación con la aplicación móvil.

Luego, se definieron las funcionalidades de cada una de las secciones anteriormente nombradas con el fin de atender las necesidades establecidas en la fase de levantamiento de requerimientos.

Funcionalidades del sitio web:

- Gestión de usuarios y tipos de usuario con el fin de ejercer control de acceso y navegación en el sitio web.
- Administración y gestión de los modelos plasmados en el modelo Entidad-Relación.
- Revisión de cada tipo de reportes con el fin de filtrar la información y generar mapas de calor basados en información confiable.
- Gestión de contenido. Para así poder mantener la sección de noticias y ayudas de la aplicación móvil actualizada.
- Construir mapas de calor a partir de la información ya filtrada de los reportes.
- Construir mapas con puntos a partir de la información ya filtrada de los reportes.
- Enviar mensajes de contacto.

Establecidas las distintas tareas que tendría el sitio web, se diseñó un par de modelos de caso en los cuales, se diagramara la interacción que tendrían el usuario y el sistema.

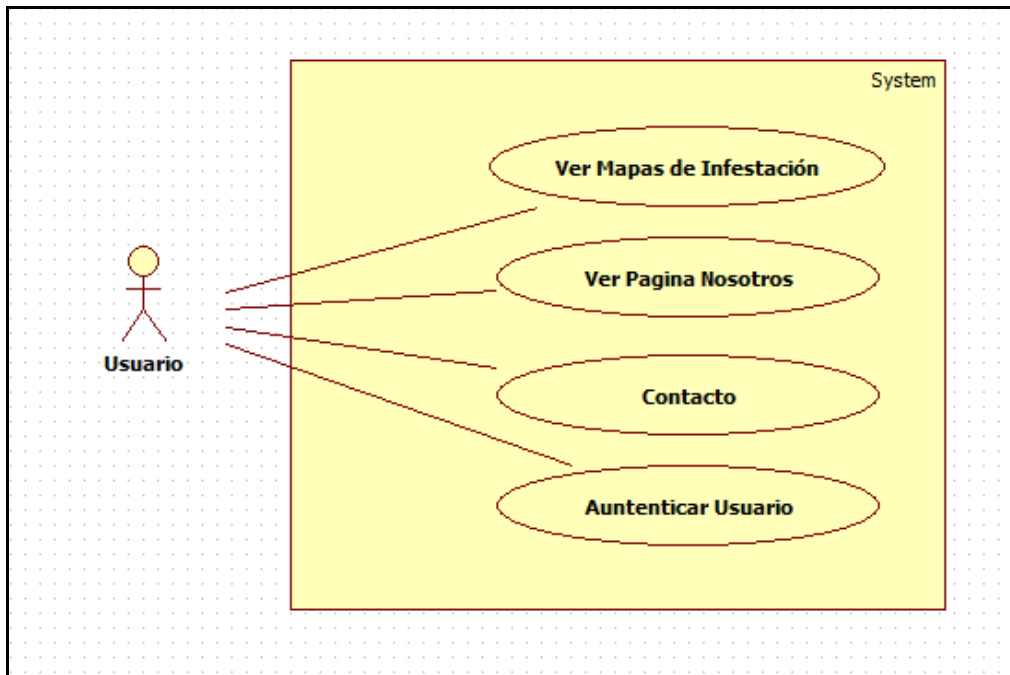


Figura 13. Caso de uso sitio web sin autenticación del usuario

Como se observa en la figura 13, el usuario sin estar autenticado solo tiene permitido ver los mapas de infestación, obtener información relacionada al proyecto en la pagina nosotros, enviar correos electrónicos a los administradores del sistema y autenticarse mediante un usuario y una contraseña.

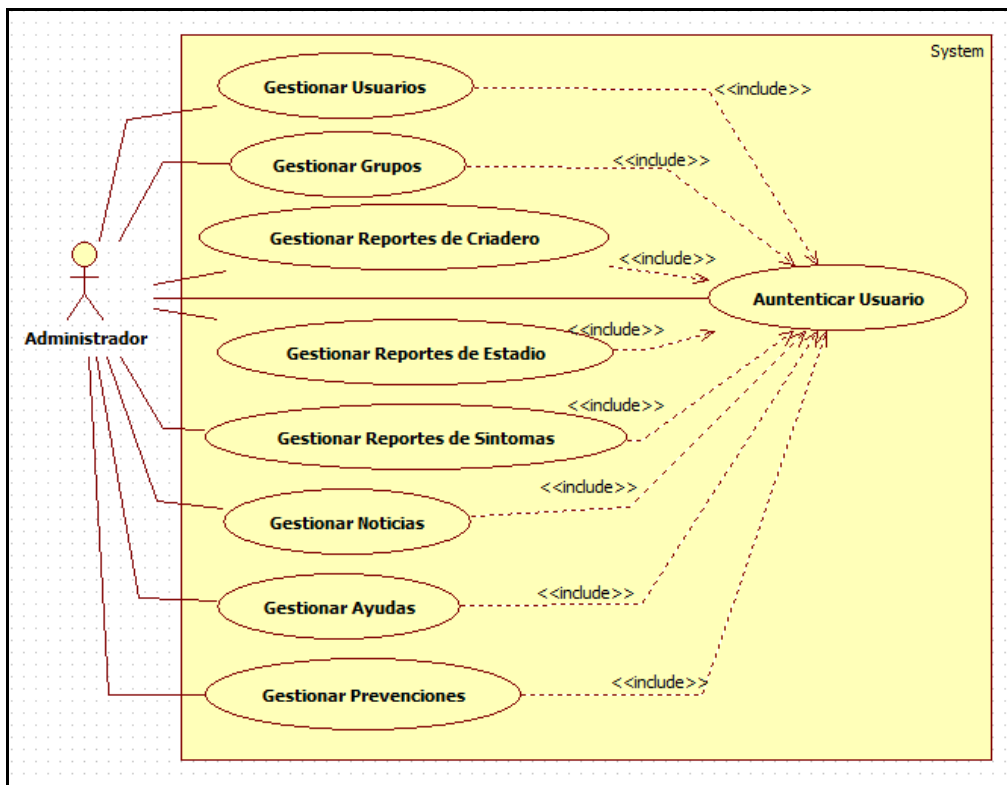


Figura 14. Caso de uso sitio web con autenticación del administrador.

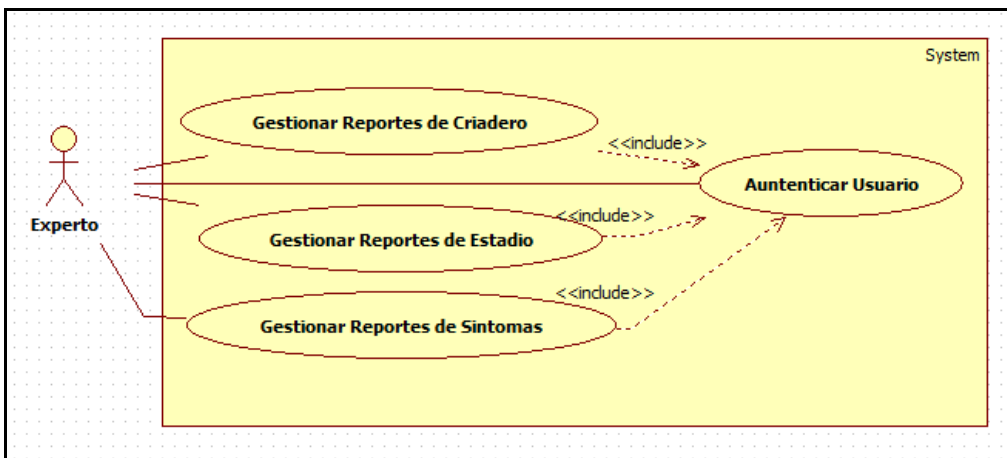


Figura 15. Caso de uso sitio web con autenticación del experto.

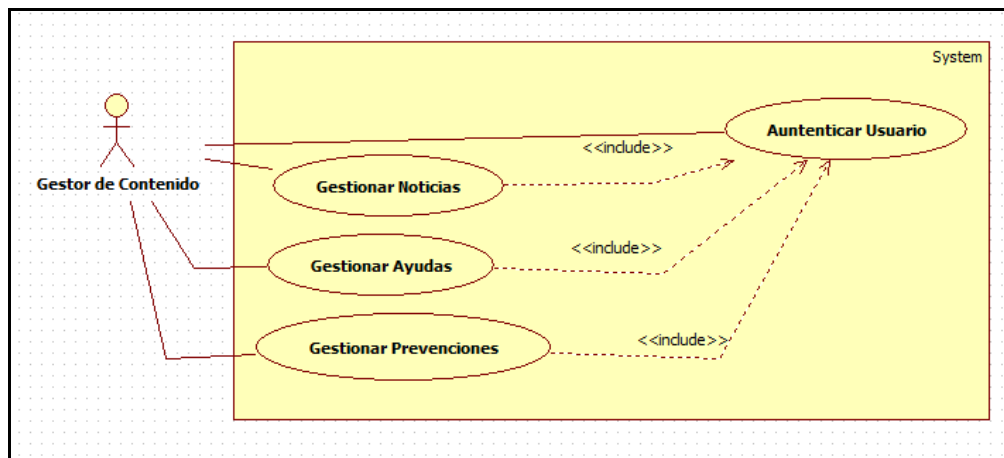


Figura 16. Caso de uso sitio web con autenticación del gestor de contenido.

En la figura 14, 15, 16, se describe como el usuario podrá interactuar con el sistema al estar autenticado. Debido a que se decidió que debían existir distintos roles mediante los cuales hacer uso del sistema, se desarrollo un caso de uso para cada tipo de rol. El administrador podrá tener acceso a todas las herramientas del sistema. El gestor de contenido solo podrá gestionar información relacionada a ayudas, prevenciones y noticias. El experto, tendrá acceso a los reportes, sin embargo, solo tiene permitido alterar la validez del reporte (es decir, marcar un reporte como valido o no valido).

Funcionalidades de la API:

- Registrar nuevos usuarios en el sistema.
- Autenticar un usuario.
- Ver información de un usuario.
- Recuperar contraseña de un usuario del sistema.
- Cambiar contraseña de un usuario del sistema.
- Ver Sección de Noticias.
- Ver Sección de Ayuda.
- Ver Sección de Prevención.
- Sección de Reportes de Criadero:
 - Ver reportes
 - Crear reportes.

- Sección de Reportes de Estadio:
 - Ver reportes
 - Crear reportes.
- Sección de Reportes de Síntoma:
 - Ver reportes
 - Crear reportes.

De la misma manera que en el diseño del sitio web, terminadas de definir las distintas funcionalidades que tendría la API se definieron algunos diagramas de casos de uso en los cuales se pudiera ver la interacción entre el usuario y el sistema.

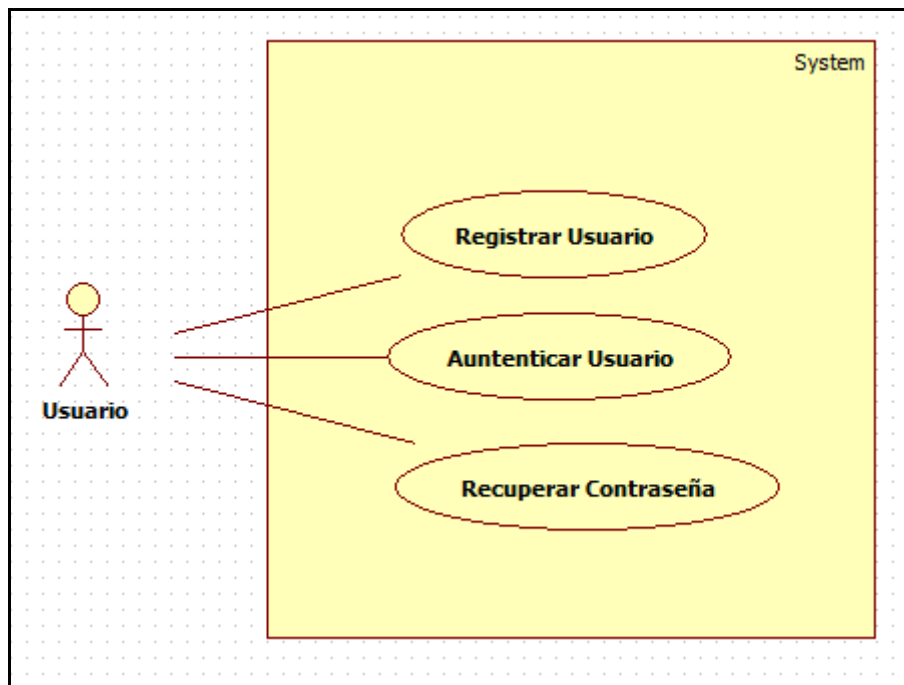


Figura 17. Caso de uso API sin autenticación del usuario.

Para el uso de la API, se definió que el usuario solo podría registrarse, autenticarse y/o recuperar su contraseña cuando no estuviera autenticado.

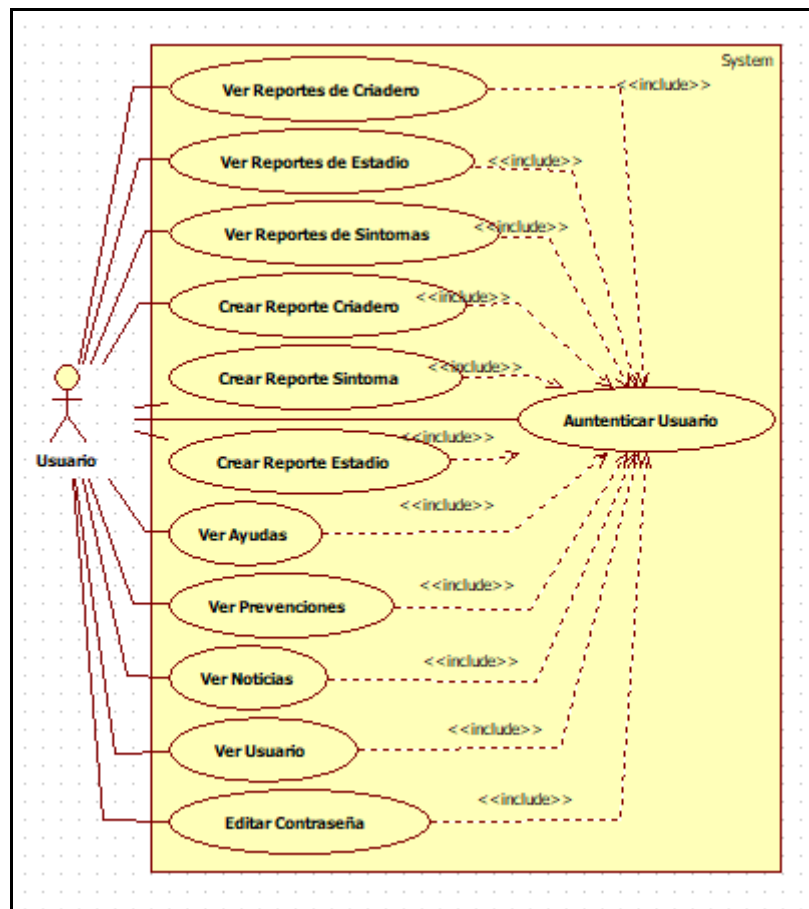


Figura 18. Caso de uso API con autenticación del usuario.

Dado el caso en que el usuario estuviera autenticado, como se observa en la figura 18, el usuario podría ver y crear reportes de criadero, síntoma y estadio, ver la información del usuario en el sistema, cambiar su contraseña y ver información relacionada a noticias prevenciones y ver ayudas.

Diseño del modelo Entidad-Relación

Definidas cada una de las tareas que debía desarrollar el sistema, se definió el modelo entidad-relación que permitiría almacenar los datos de la solución como se observa a continuación, en la figura 19:

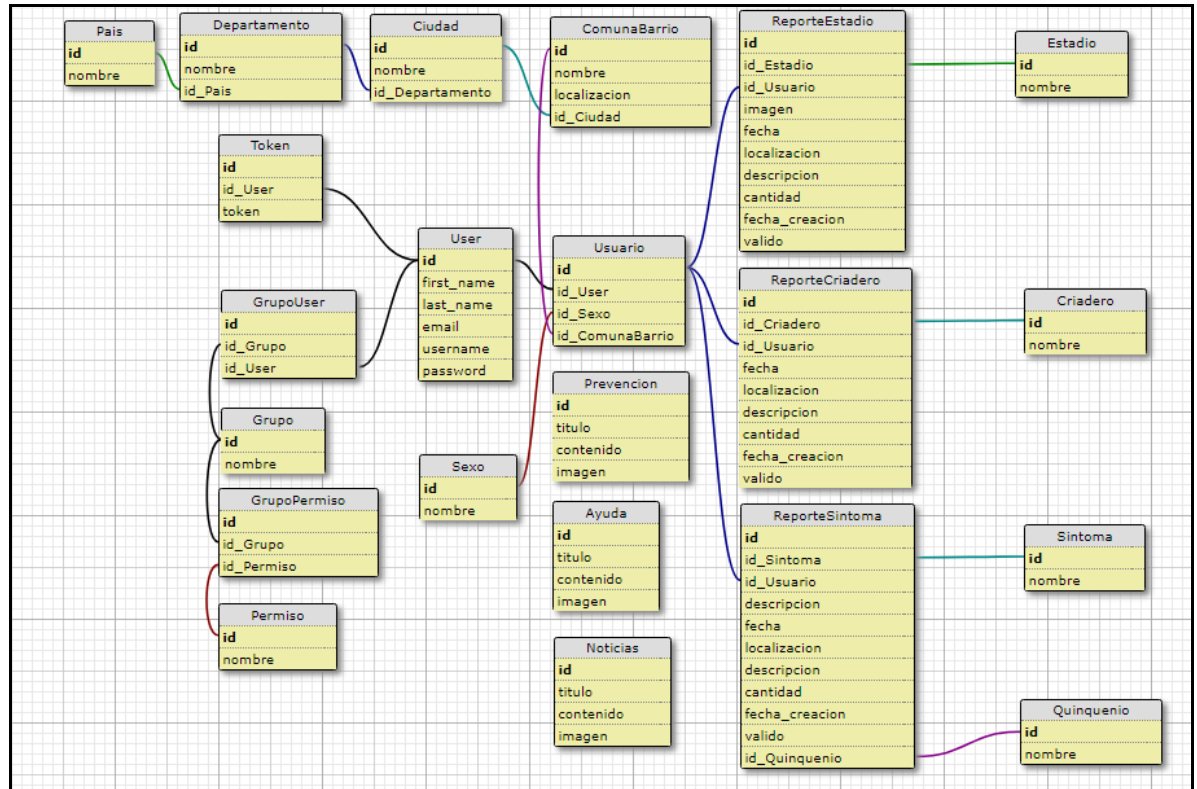


Figura 19. Modelo entidad-relación de la solución.

El modelo entidad-relación está compuesto por 22 tablas, la razón por la cual fueron construidas se describe a continuación:

- Tabla User y tabla Usuario:

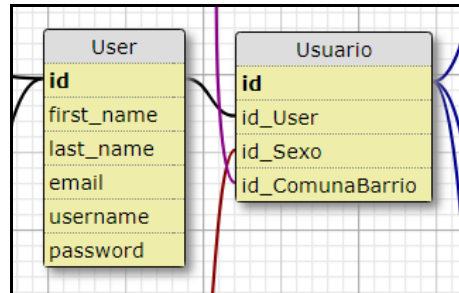


Figura 20. Tablas User y Usuario del modelo entidad-relación.

Este par de tablas está constituido de esta manera debido a que Django por defecto ya cuenta con un sistema de autenticación y permisos entre los que está inmersa la tabla User y como buena práctica, el marco de trabajo considera que si el desarrollador necesita más campos de los contenidos en la tabla User, se debe crear una nueva tabla, que tenga por referencia a la tabla User, debido a ello se creó la tabla Usuario y en dicha tabla un campo tiene como llave foránea el campo id de la tabla User.

- Tablas Prevención, Noticia y Ayuda:

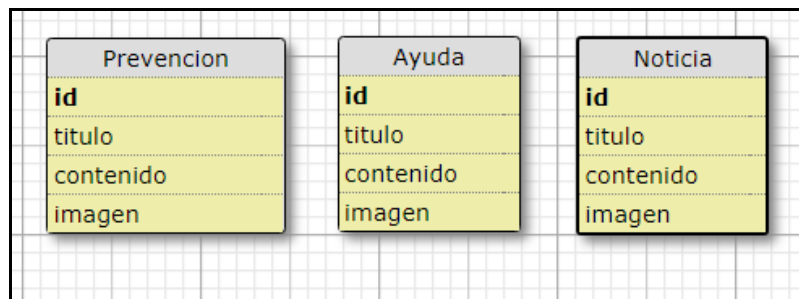


Figura 21. Tablas Prevencion, Ayuda y Noticia del modelo entidad-relación.

Estas tres tablas fueron diseñadas debido a que el proyecto se comunica con una aplicación móvil y se consideró necesario que ella pudiera brindar información relacionada a cómo prevenir el vector y de compartir noticias relacionadas al mismo a los usuarios de dicha aplicación.

- Tablas Grupo, Permiso, GrupoPermiso y GrupoUser:

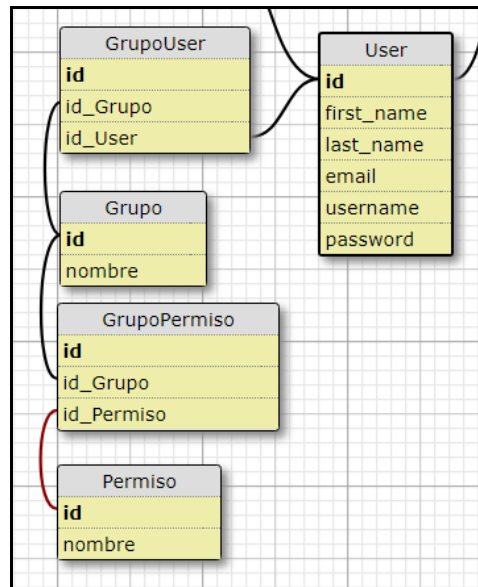


Figura 22. Tablas Grupo, Permiso, User, GrupoPermiso y GrupoUser en modelo entidad-relación.

Estas tablas están definidas por el marco de trabajo Django y permiten crear grupos en los cuales los usuarios pertenecientes a dichos grupos podrán tener ciertos permisos de gestión del sistema. Estas tablas, fueron de gran utilidad debido a que permitió que en el sistema se pudieran definir roles (representados por grupos) y así delimitar las capacidades que tenían los distintos usuarios para hacer gestión de la información. Para este caso, se definieron dos distintos roles: Gestores de Contenido y Expertos.

- Tablas Pais, Departamento, Ciudad, ComunaBarrio:

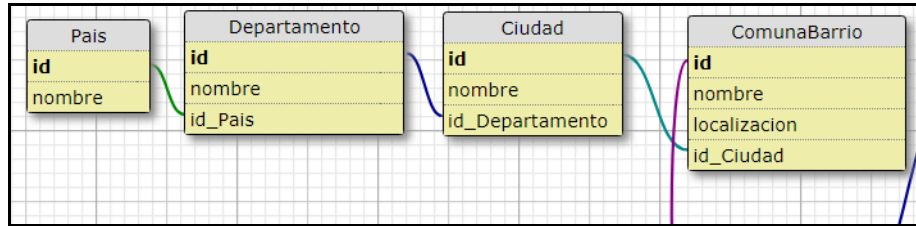


Figura 23. Tablas Pais, Departamento, Ciudad, ComunaBarrio en modelo entidad-relación.

Estas tablas permiten gestionar en el sistema ciudades, departamentos, países y comunas o barrios. Son de utilidad debido a que el usuario contiene una llave foránea en la que deberá indicar la comuna o el barrio en el que vive.

- Tabla Token:

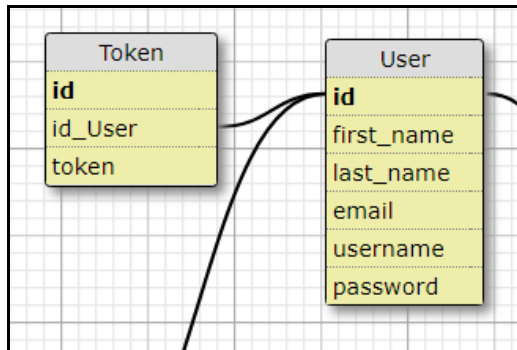


Figura 24. Tabla Token y su relación con la tabla User en el modelo Entidad-Relación.

Esta tabla se creó con el fin de autenticar los usuarios de la API mediante el uso de token. De esta manera, cuando un usuario inicie sesión en la API, obtendrá un token y al momento de realizar peticiones al servidor el se identificará mediante dicho token.

- Tablas ReporteEstadio, ReporteCriadero, ReporteSintoma, Estadio, Criadero, Sintoma y Quinquenio:

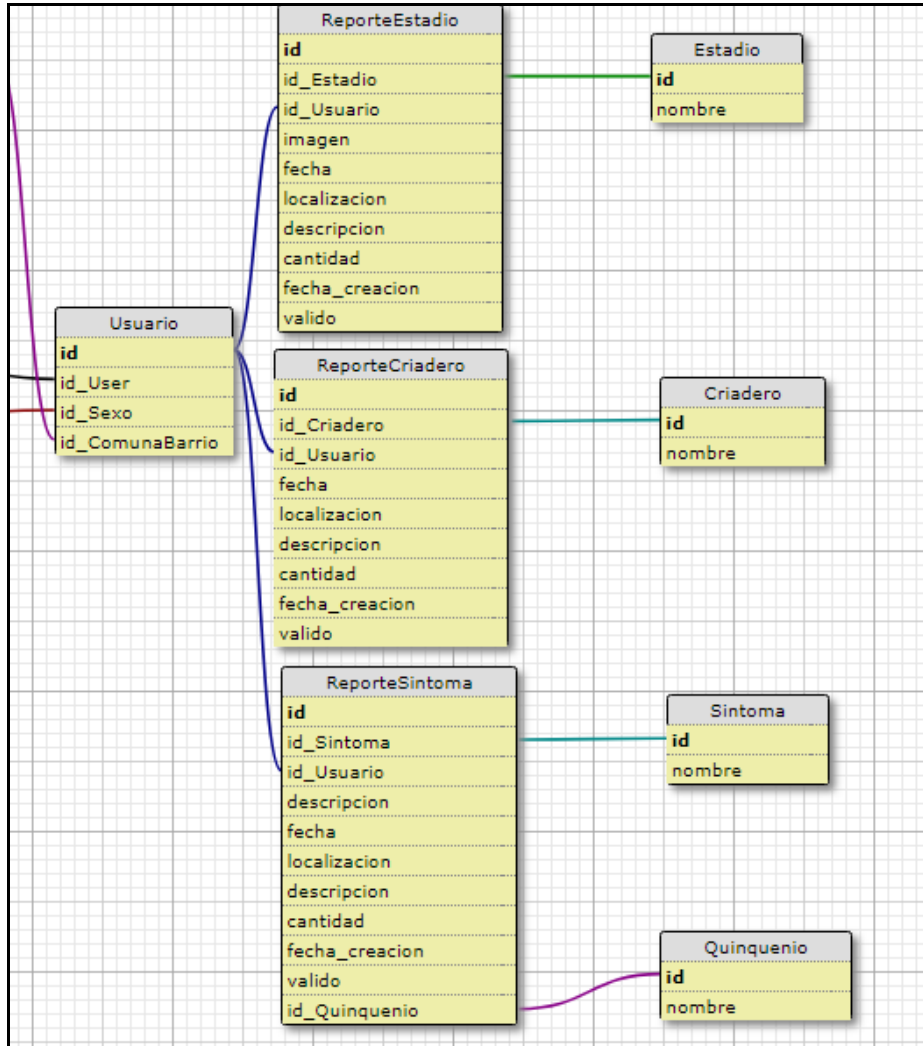


Figura 25. Tablas de Reporte, Estadio, Criadero, Síntoma y quinquenio en el modelo entidad-relación.

Estas tablas almacenan la información crítica del sistema, los reportes. Para los reportes, se decidió hacer tres tablas en las cuales se almacenarían tres distintos tipos de reporte: de criadero, de estadio del vector y de un síntoma que padeciera una persona. Debido a que existían distintos síntomas, tipos de criadero y estadios del vector *Aedes Aegypti*, se creó una tabla respectivamente para síntomas, criaderos y estadios. Para terminar, en reporte síntoma se relacionó la tabla a un quinquenio, con el fin de poder filtrar los reportes de síntoma según la edad de quien lo padeciera.

3. Implementación de la solución.

Al tener claras las tareas que desarrollaría el sistema y el modelo de datos sobre el cual se trabajaría, se decidió iniciar la implementación de la solución. Para el ambiente de desarrollo se decidió que debía hacerse uso del sistema operativo Debian 8 Jessie, esto debido a que el servidor en el cual se desplegaría la solución hacía uso de este mismo sistema operativo y se buscaba que el entorno de desarrollo fuera lo mas cercano al ambiente de producción. Posterior a ello, se realizó la instalación de las herramientas necesarias para el desarrollo de la solución en el ambiente de desarrollo.

Terminada la instalación de dichas herramientas, se empezó a trabajar sobre el código Python que determinaría el funcionamiento de todo el sistema. Para iniciar, se modeló cada una de las entidades que habían sido plasmadas en el modelo entidad-relación. En la figura 26 se observa cómo es el código Python que describe uno de los modelos:

```
class ReportarCriadero(models.Model):
    criadero_opciones = (
        ('llanta', 'Llanta'), ('tanque', 'Tanque'), ('lavadero', 'Lavadero'),
        ('florero', 'Florero'), ('recipiente', 'Recipiente'), ('botella', 'Botella'),
        ('otro', 'Otro')
    )
    usuario_creador = models.ForeignKey(User)
    criadero = models.CharField(max_length=11, choices=criadero_opciones)
    fecha = models.DateTimeField(verbose_name="Fecha en la que se tomó el reporte")
    localizacion = models.PointField()
    descripcion = models.TextField(blank=True, null=True)
    cantidad = models.IntegerField()
    fecha_creacion = models.DateTimeField(auto_now_add=True, verbose_name="Fecha en la que se almacena el reporte")
    valido = models.NullBooleanField(choices=valido_choices, default=None, verbose_name="Este reporte es válido?")
```

Figura 26. Código Python del modelo ReportarCriadero

Para cada tabla del modelo entidad-relación se hizo la misma tarea: mediante una clase, codificar cada uno de los campos que tenían las tablas, a excepción, de los modelos de Grupo, Permiso, GrupoPermiso, GrupoUser, Token y User. Ya que dichos modelos, son definidos por el marco de trabajo a la hora de crear el proyecto.

Terminada la tarea de modelar cada uno de los modelos planteados en el modelo entidad relación, se procedió a crear algo denominado por Django como “sitio administrativo”. Este sitio administrativo es el lugar al cual los administradores, expertos y gestores de contenido ingresan para gestionar información del sistema.

A continuación, un ejemplo de la manera en que se codifica un modelo en el sitio administrativo de Django:

```
@admin.register(Noticia)
class NoticiaAdmin(admin.ModelAdmin):
    search_fields = ('titulo',)
    list_display = ('titulo', 'contenido', 'fecha_creacion',)
```

Figura 27. Registro del Modelo Noticia en el sitio administrativo.

Para cada modelo creado, se realizó esta tarea, con el fin de que la información contenida en el sistema fuera manipulable.

En este punto, el sistema ya podía ser puesto en funcionamiento. Hasta este punto solo podrían hacer uso los administradores del sistema, las funcionalidades a las que tenían acceso eran:

- Autenticarse mediante una página de inicio de sesión.

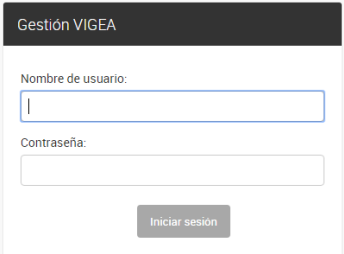


Figura 28. Autenticación en sistema de administración.

- Gestionar los datos del sistema a través del administrador (El administrador permite ver, crear, editar y eliminar cualquier información que ha sido almacenada).

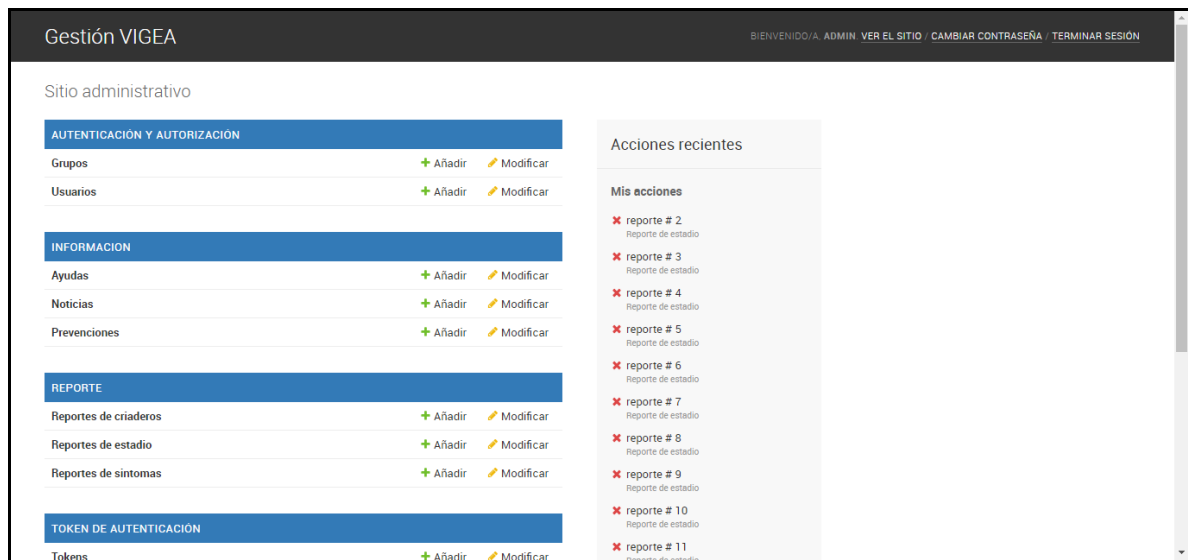


Figura 29. Administrador de Django en este proyecto.

Teniendo el administrador listo, se considero necesario proceder a desarrollar la API que permitiría la comunicación entre el sistema Web y la aplicación móvil. Para la aplicación móvil se desarrollaron: Serializadores y APIView. Para este proyecto, el desarrollo de la API se hizo usando Django Rest Framework, por lo que podríamos definir un Serializador como un tipo de formulario que nos permitiría diligenciar la información solicitada por los distintos Web Services. En la figura 30 se observa uno de los serializadores de la API:

```
class ReportarEstadioSerializer(Serializer):

    usuario_creador = serializers.IntegerField()
    imagen = serializers.CharField()
    fecha = serializers.DateTimeField(format="%Y-%m-%d %H:%M") #EJEMPLO:"2018-(1-12)-(1-31) (1-23):(0-59)"
    localizacion_x = serializers.FloatField()
    localizacion_y = serializers.FloatField()
    descripcion = serializers.CharField(required=False)
    cantidad = serializers.IntegerField()
    estadio = serializers.CharField(max_length=8) #larva, pupa, zancudo
```

Figura 30. Código del Serializador usado para hacer un Reporte de Estadio

El Serializador además de permitir diligenciar la información que enviamos al Web Service, también se encarga de revisar si los datos ingresados en los distintos campos son del tipo de dato necesario. Por ejemplo, si enviásemos un String en un campo que debe ser de tipo Integer, el serializador le enviara a la APIView una lista con los errores hallados en los datos ingresados al serializer. De la misma

manera, en caso de que la revisión sea correcta, permitirá acceder a los datos y procesarlos según se desee.

Terminados cada uno de los Web Services planeados para la API, se paso de nuevo al sitio web, para este caso, se empezaron a crear las vistas que permitirían ver los mapas de infestación, la pagina con información del proyecto y la pagina con el formulario de contacto.

Para la visualización de los mapas de calor en la web, el proyecto necesitaba hacer uso de Tomcat, Geoserver y Openlayers (adicional a Django que había sido lo que se había trabajado hasta ese punto del proyecto).

Para empezar, se realizó la instalación de Tomcat, luego se realizo el despliegue de GeoServer. Dichas herramientas se veían así:

The Apache Software Foundation http://www.apache.org/				
Tomcat Web Application Manager				
Message: OK				
Manager				
List Applications	HTML Manager Help	Manager Help	Server Status	
Applications				
Path	Display Name	Running	Sessions	Commands
/	Welcome to Tomcat	true	0	Start Stop Reload Undeploy <input type="button" value="Expire sessions"/> with idle ≥ 30 minutes
/docs	Tomcat Documentation	true	0	Start Stop Reload Undeploy <input type="button" value="Expire sessions"/> with idle ≥ 30 minutes

Figura 31. Manager App de Tomcat

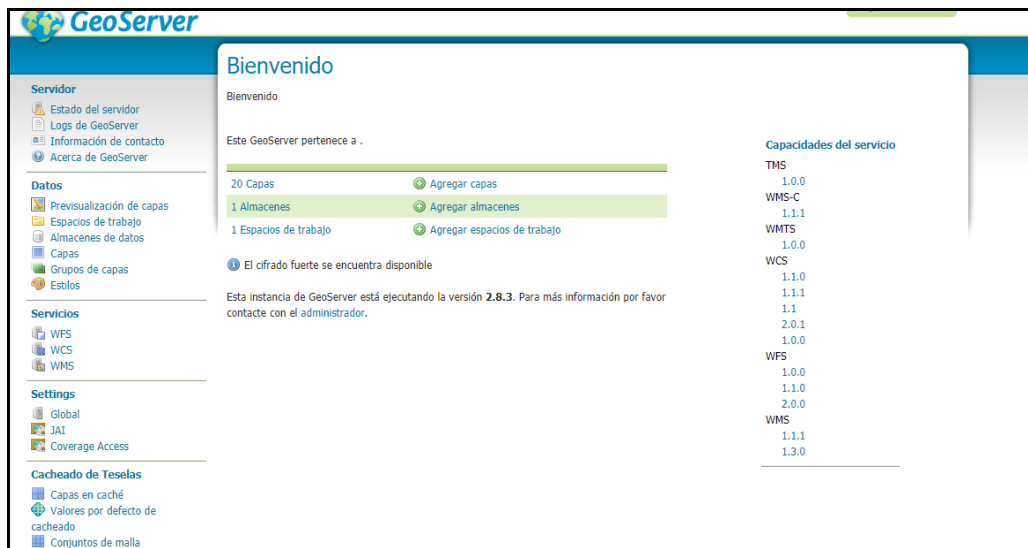


Figura 32. Administrador de GeoServer.

Pudiendo hacer uso de GeoServer, se creo un espacio de trabajo, luego se creo algo denominado por GeoServer como almacén de datos, que consiste en la configuración de conexión a una base de datos y además la parametrización de dicha conexión. Al tener el almacén de datos se empezaron a configurar capas, con el fin de que estas permitieran la lectura de los datos guardados en reportes, a través de un formato GeoJson. Teniendo dichas capas, ahora se tendría la capacidad de servir dichas capas para ser consumidas por OpenLayers. La siguiente figura es la visualización de una capa de GeoServer en formato GeoJson:

```
{
  "type": "FeatureCollection",
  "totalFeatures": 11,
  "features": [
    {
      "type": "Feature",
      "id": "reporteestadio_larva.57",
      "geometry": {
        "type": "Point",
        "coordinates": [
          4.14, -73.13
        ]
      },
      "geometry_name": "localizacion",
      "properties": {
        "imagen": "reporte_estadio/2018/05/temp.jpeg",
        "fecha": "2018-05-08T14:18:00Z",
        "descripcion": "algo",
        "cantidad": 10,
        "estadio": "larva",
        "fecha_creacion": "2018-05-08T14:18:50.817Z",
        "usuario_creador_id": 1,
        "valido": true,
        "bbox": [
          4.14, -73.13, 4.14, -73.13
        ]
      }
    },
    {
      "type": "Feature",
      "id": "reporteestadio_larva.51",
      "geometry": {
        "type": "Point",
        "coordinates": [
          -73.6435932, 4.1465178
        ]
      },
      "geometry_name": "localizacion",
      "properties": {
        "imagen": "temp_PuOVVvg.png",
        "fecha": "2018-04-19T19:31:57Z",
        "descripcion": "N/A",
        "cantidad": 1,
        "estadio": "larva",
        "fecha_creacion": "2018-04-19T19:31:52.140Z",
        "usuario_creador_id": 59,
        "valido": true,
        "bbox": [
          -73.6435932, 4.1465178, -73.6435932, 4.1465178
        ]
      }
    },
    {
      "type": "Point",
      "coordinates": [
        -73.6435439, 4.1464796
      ]
    },
    {
      "type": "Feature",
      "id": "reporteestadio_larva.50",
      "geometry": {
        "type": "Point",
        "coordinates": [
          -73.6435439, 4.1464796
        ]
      },
      "geometry_name": "localizacion",
      "properties": {
        "imagen": "temp_alicJH05.png",
        "fecha": "2018-04-19T19:26:35Z",
        "descripcion": "N/A",
        "cantidad": 12,
        "estadio": "larva",
        "fecha_creacion": "2018-04-19T19:31:35.464Z",
        "usuario_creador_id": 63,
        "valido": true,
        "bbox": [
          -73.6435439, 4.1464796, -73.6435439, 4.1464796
        ]
      }
    },
    {
      "type": "Feature",
      "id": "reporteestadio_larva.47",
      "geometry": {
        "type": "Point",
        "coordinates": [
          -73.6429717, 4.1462257
        ]
      },
      "geometry_name": "localizacion",
      "properties": {
        "imagen": "temp_lkmqK7e.png",
        "fecha": "2018-04-15T12:48:47Z",
        "descripcion": "N/A",
        "cantidad": 10,
        "estadio": "larva",
        "fecha_creacion": "2018-04-15T12:48:35.592Z",
        "usuario_creador_id": 59,
        "valido": true,
        "bbox": [
          -73.6429717, 4.1462257, -73.6429717, 4.1462257
        ]
      }
    },
    {
      "type": "Feature",
      "id": "reporteestadio_larva.40",
      "geometry": {
        "type": "Point",
        "coordinates": [
          -73.6414622, 4.1587317
        ]
      },
      "geometry_name": "localizacion",
      "properties": {
        "imagen": "temp_dk6mRa.png",
        "fecha": "2018-04-12T03:00:06Z",
        "descripcion": "N/A",
        "cantidad": 3,
        "estadio": "larva",
        "fecha_creacion": "2018-04-12T03:00:22.323Z",
        "usuario_creador_id": 57,
        "valido": true,
        "bbox": [
          -73.6414622, 4.1587317, -73.6414622, 4.1587317
        ]
      }
    },
    {
      "type": "Feature",
      "id": "reporteestadio_larva.21",
      "geometry": {
        "type": "Point",
        "coordinates": [
          -73.5838609, 4.0742764
        ]
      },
      "geometry_name": "localizacion",
      "properties": {
        "imagen": "temp_iEe0Nxb.png",
        "fecha": "2018-04-05T19:28:50Z",
        "descripcion": "N/A",
        "cantidad": 1,
        "estadio": "larva",
        "fecha_creacion": "2018-04-05T19:29:06.549Z",
        "usuario_creador_id": 45,
        "valido": true,
        "bbox": [
          -73.5838609, 4.0742764, -73.5838609, 4.0742764
        ]
      }
    },
    {
      "type": "Feature",
      "id": "reporteestadio_larva.17",
      "geometry": {
        "type": "Point",
        "coordinates": [
          -73.5612221, 4.0863296
        ]
      },
      "geometry_name": "localizacion",
      "properties": {
        "imagen": "temp_21V6coh.png",
        "fecha": "2018-04-05T18:02:07Z",
        "descripcion": "N/A",
        "cantidad": 1,
        "estadio": "larva",
        "fecha_creacion": "2018-04-05T18:01:55.809Z",
        "usuario_creador_id": 8,
        "valido": true,
        "bbox": [
          -73.5612221, 4.0863296, -73.5612221, 4.0863296
        ]
      }
    },
    {
      "type": "Feature",
      "id": "reporteestadio_larva.16",
      "geometry": {
        "type": "Point",
        "coordinates": [
          -78.9944428125, 8.41278
        ]
      },
      "geometry_name": "localizacion",
      "properties": {
        "imagen": "temp_emporio.png",
        "fecha": "2018-03-22T02:09:37Z",
        "descripcion": "N/A",
        "cantidad": 1,
        "estadio": "larva",
        "fecha_creacion": "2018-03-22T02:09:54.736Z",
        "usuario_creador_id": 1,
        "valido": true,
        "bbox": [
          -78.9944428125, 8.41278, -78.9944428125, 8.41278
        ]
      }
    },
    {
      "type": "Feature",
      "id": "reporteestadio_larva.67",
      "geometry": {
        "type": "Point",
        "coordinates": [
          -73.6432909, 4.1465299
        ]
      },
      "geometry_name": "localizacion",
      "properties": {
        "imagen": "reporte_estadio/2018/05/temp_kj9Vyc3.png",
        "fecha": "2018-05-17T19:39:35Z",
        "descripcion": "N/A",
        "cantidad": 1,
        "estadio": "larva",
        "fecha_creacion": "2018-05-17T19:39:07.453Z",
        "usuario_creador_id": 93,
        "valido": true,
        "bbox": [
          -73.6432909, 4.1465299, -73.6432909, 4.1465299
        ]
      }
    },
    {
      "type": "Feature",
      "id": "reporteestadio_larva.14",
      "geometry": {
        "type": "Point",
        "coordinates": [
          -73.6412742, 4.1585303
        ]
      },
      "geometry_name": "localizacion",
      "properties": {
        "imagen": "temp_P0arvt2.png",
        "fecha": "2018-03-01T02:06:29Z",
        "descripcion": "N/A",
        "cantidad": 3,
        "estadio": "larva",
        "fecha_creacion": "2018-03-01T02:07:03.159Z",
        "usuario_creador_id": 7,
        "valido": true,
        "bbox": [
          -73.6412742, 4.1585303, -73.6412742, 4.1585303
        ]
      }
    },
    {
      "type": "Feature",
      "id": "reporteestadio_larva.1",
      "geometry": {
        "type": "Point",
        "coordinates": [
          -73.627187, 4.11818
        ]
      },
      "geometry_name": "localizacion",
      "properties": {
        "imagen": "temp.png",
        "fecha": "2018-02-13T16:58:00Z",
        "descripcion": "probando web service",
        "cantidad": 1,
        "estadio": "larva",
        "fecha_creacion": "2018-02-13T17:00:00.053Z",
        "usuario_creador_id": 1,
        "valido": true,
        "bbox": [
          -73.627187, 4.11818, -73.627187, 4.11818
        ]
      }
    }
  ],
  "crs": {
    "type": "name",
    "properties": {
      "name": "urn:ogc:def:crs:EPSG:4326",
      "bbox": [
        -78.9944428125, -73.13, 4.14, 8.41278
      ]
    }
  }
}
```

Figura 33. Visualización de una capa de GeoServer en formato GeoJson.

Al tener acceso a las capas de GeoServer, se debía ahora hacer la configuración de OpenLayers con el fin de consumir dichas capas. Para ello, lo primero que se

hizo fue añadir la librería a los estáticos del proyecto, posteriormente, se configuro de tal manera que se ofreciera un mapa de calor por cada subtipo de reporte, por ejemplo: para los reportes de estadio, se ofrecería un mapa de infestación general y otros 3 mapas de infestación, uno para el estadio larva, otro para el estadio pupa y otro para el estadio zancudo, esto mismo se aplico con los otros dos tipos de reportes, a continuación, algunos segmentos de código de la configuración del OpenLayers y la explicación de dicho código:

```
mapas.push(  
  new ol.layer.Tile  
  ({  
    title: 'Open Street View',  
    type: 'base',  
    visible: true,  
    source: new ol.source.OSM()  
  })  
);  
var groupMapas = new ol.layer.Group({  
  title: 'Mapas Base',  
  layers: mapas  
});
```

Figura 34. Declaración de Open Street Map como mapa base para el mapa de infestación.

Lo primero que se realizo fue definir los mapas bases sobre los cuales se visualizarían los mapas de infestación. Para este proyecto, se hizo uso del mapa de Open Street Map y el mapa de Bing.

```
criaderos.push(  
  new ol.layer.Heatmap  
  ({  
    title: 'Otro',  
    visible: false,  
    source: new ol.source.Vector({  
      format: new ol.format.GeoJSON(),  
      url: 'http://190.60.95.13:8080/geoserver/webgis_aedes/ows?service=WFS&version=1.0.0&request=GetFeature&' +  
        'typeName=webgis_aedes:reportarcriadero_otro&maxFeatures=50&outputFormat=application%2Fjson'  
    })),  
    weight: function (feature) {  
      var weightProperty = feature.get('cantidad');  
      // perform some calculation to get weightProperty between 0 - 1  
      weightProperty = weightProperty / max_otro; // this was your suggestion - make sure this makes sense  
      return weightProperty;  
    }  
  })  
);
```

Figura 35. Configuración de la visualización de una capa en OpenLayers.

Luego, se procedió a configurar las distintas capas mediante los que se visualizarían los mapas de calor, como se observa en la figura 35, a la hora de declarar una capa, se debía indicar su nombre, si seria visible al acceder al mapa, el origen de los datos y en weight se definían la escala de color en el mapa de

calor que debían tomar cada uno de los puntos. Para el caso de reporte de estadio y de criadero, el weight se vio definido por la cantidad del reporte, por ejemplo, un reporte de criadero con diez llantas debía mostrar un color más fuerte que un reporte de criadero con una llanta. En las siguientes figuras se muestra el resultado de la configuración de los mapas de infestación:

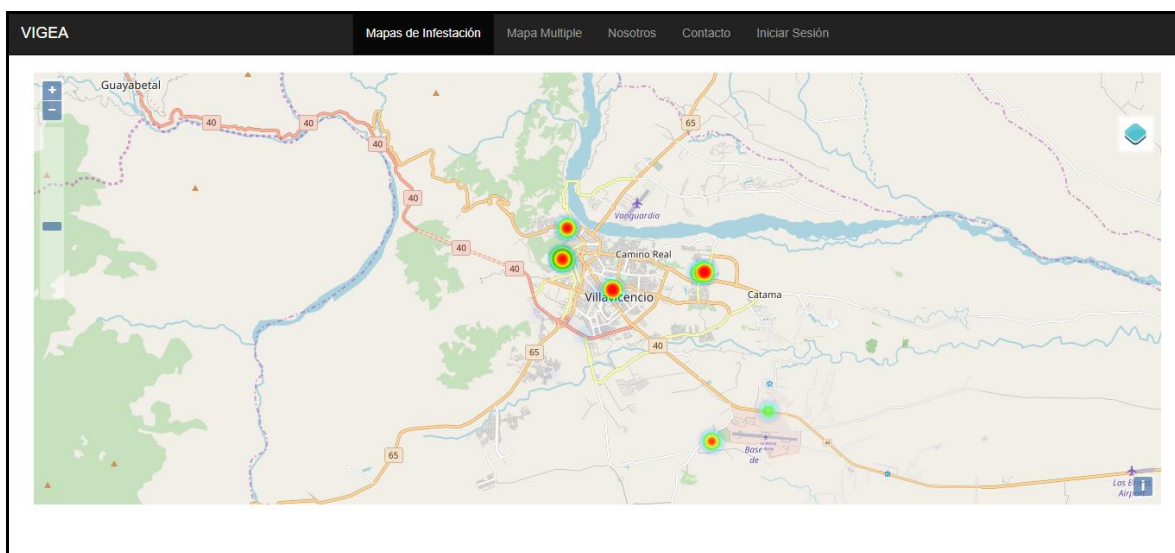


Figura 36. Mapa de infestación de todos los reportes con mapa base de Open Street Map.

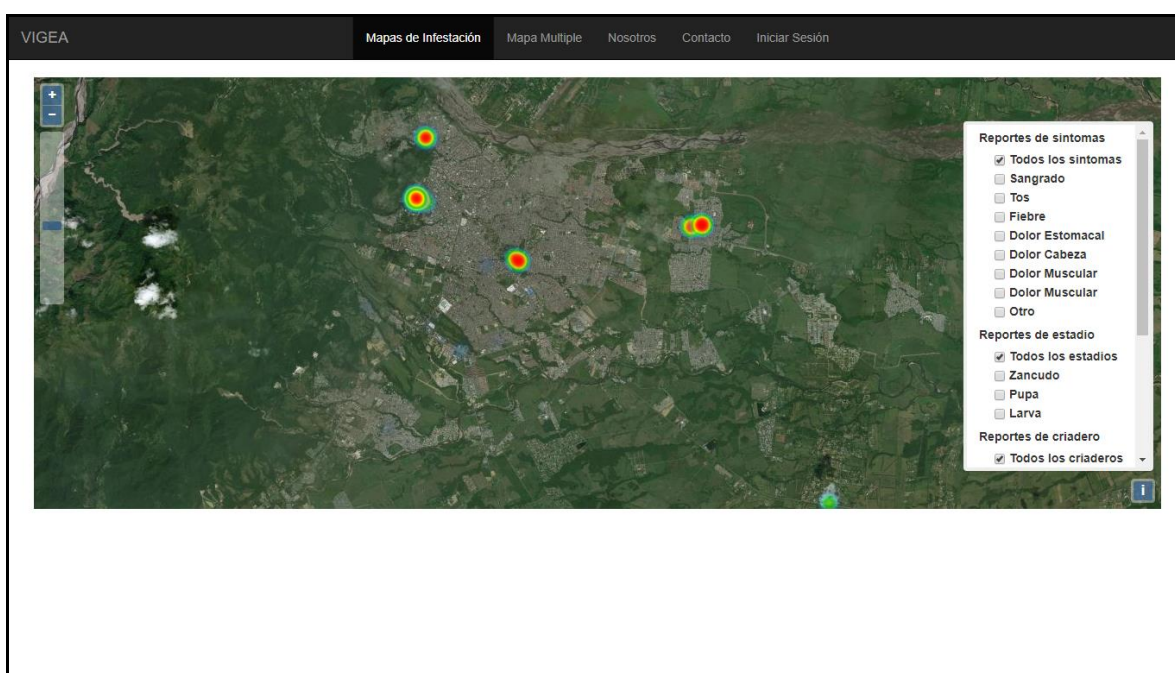


Figura 37. Mapas de infestación de todos los reportes con mapa base de Bing.

Al terminar esto, ya contábamos con un sistema funcional que cumplía con todas las tareas que se habían definido en la planeación.

Ahora, todo lo que se había hecho hasta ese momento, debía ser replicado en el servidor de producción para acceder al proyecto mediante la Web.

Tal como se había definido, la aplicación encargada de funcionar como servidor web fue Nginx debido a que era el mas recomendado por el marco de trabajo, para ver detalles del despliegue del servidor en producción revisar el **anexo del manual técnico**, en el, se explica minuciosamente cada uno de los pasos que se llevaron a cabo para poner este proyecto en producción.

Puesto en producción el proyecto, se procedió a crear en producción los roles necesarios para la administración del sistema, pues como se había contemplado al inicio: era necesario contar con distintos roles para controlar el acceso al sistema. Con el fin de dar solución a este requerimiento se hizo uso del sistema de grupos y permisos definido ya por el marco de trabajo Django, a continuación, se observan los distintos formularios mediante los cuales se definen estos grupos y permisos:

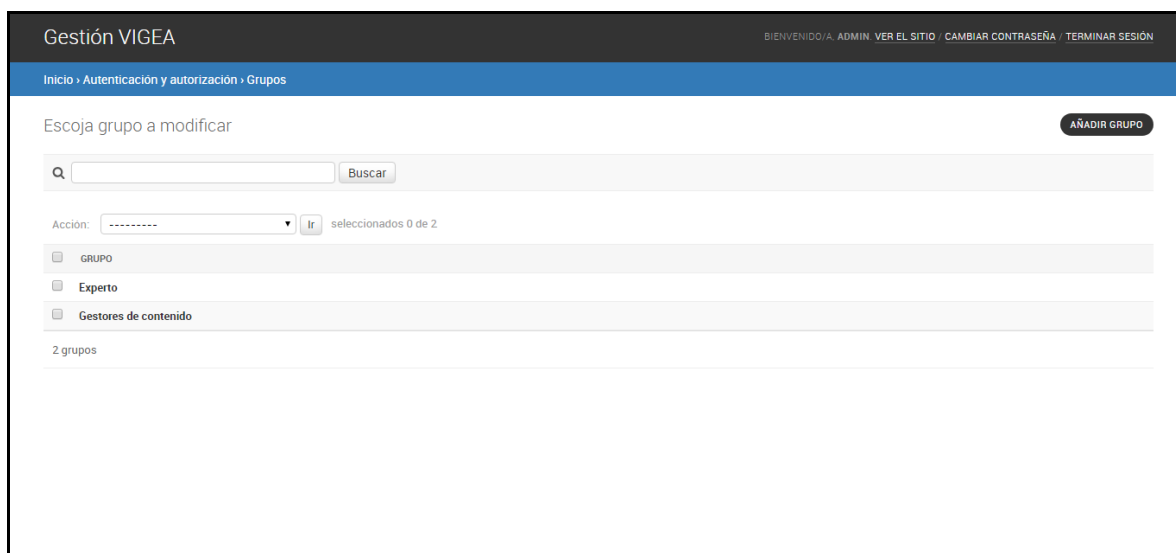


Figura 38. Lista de grupos que existen en el sistema web.

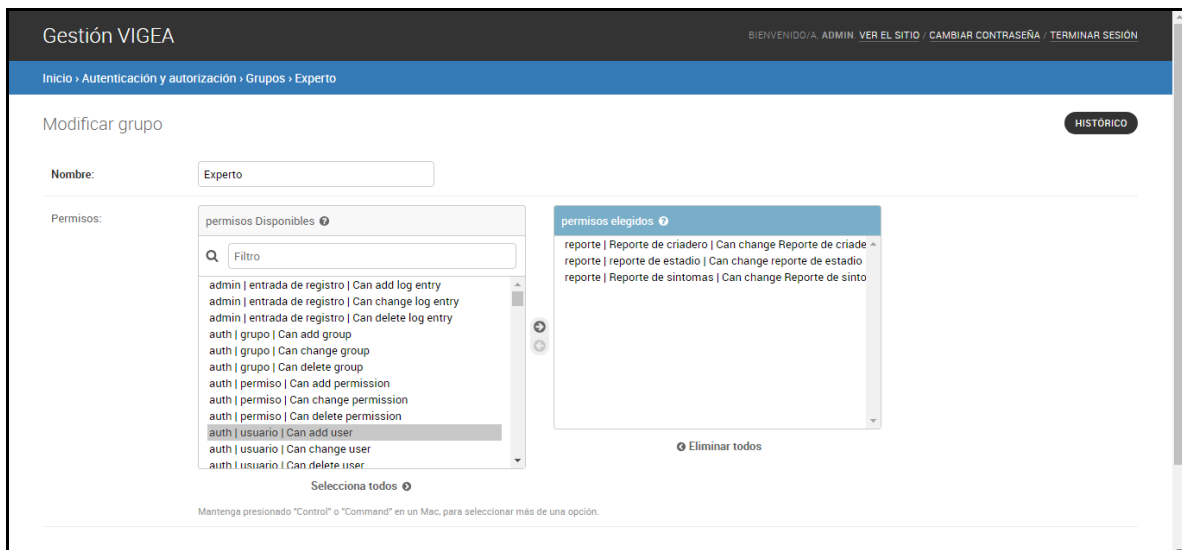


Figura 39. Gestión de un grupo al interior del sistema web. En la figura se observa que se puede editar el nombre del grupo y las tablas de las cuales puede hacer gestión dicho grupo.

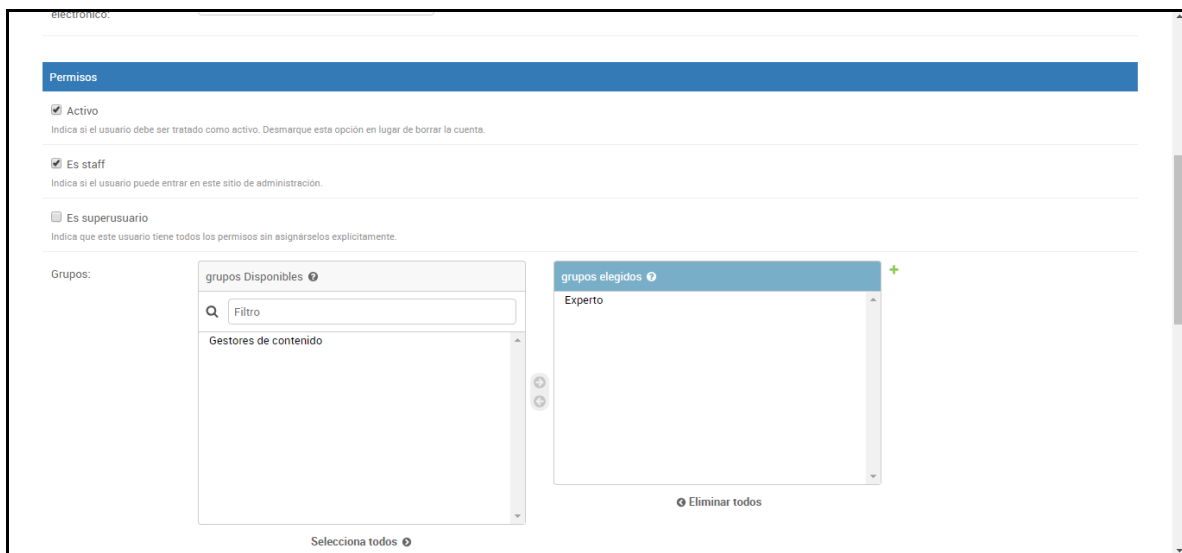


Figura 40. Se observa un formulario en el cual se pueden editar los niveles de acceso al sistema web. Para acceder al sistema de administrativo, se le dio permisos de staff y se le añadió al grupo de Experto.

Concluido el ultimo requerimiento se considero que el desarrollo estaba completo y que se debía iniciar una fase de pruebas.

4. Pruebas.

Para la sección de pruebas, se decidió compartir con los estudiantes del curso de epidemiología del programa de Enfermería de la Universidad de los Llanos las experiencias obtenidas hasta ese momento en el desarrollo del proyecto, se les socializo el estado del proyecto y la forma en el cual este funcionaba, se les solicito la instalación de la aplicación móvil y el uso de esta. Al término de la socialización se hallaron un grupo de errores y un par de elementos que a consideración de los estudiantes y de la profesora del curso eran necesarios implementar o quitar del proyecto, por lo que sus criticas fueron acogidas y se procedió a la respectiva corrección de errores.



Figura 41. Fotografía de socialización del proyecto a los estudiantes del curso epidemiología del programa de Enfermería de la Universidad de los Llanos.

5. Corrección de errores.

Para la corrección de errores, se corrigió un error que provocaba que a la hora de ingresar al detalle de un reporte, los expertos no pudieran visualizar la información

contenida por dicho reporte, por lo que, el reporte se visualizaba de la siguiente manera:

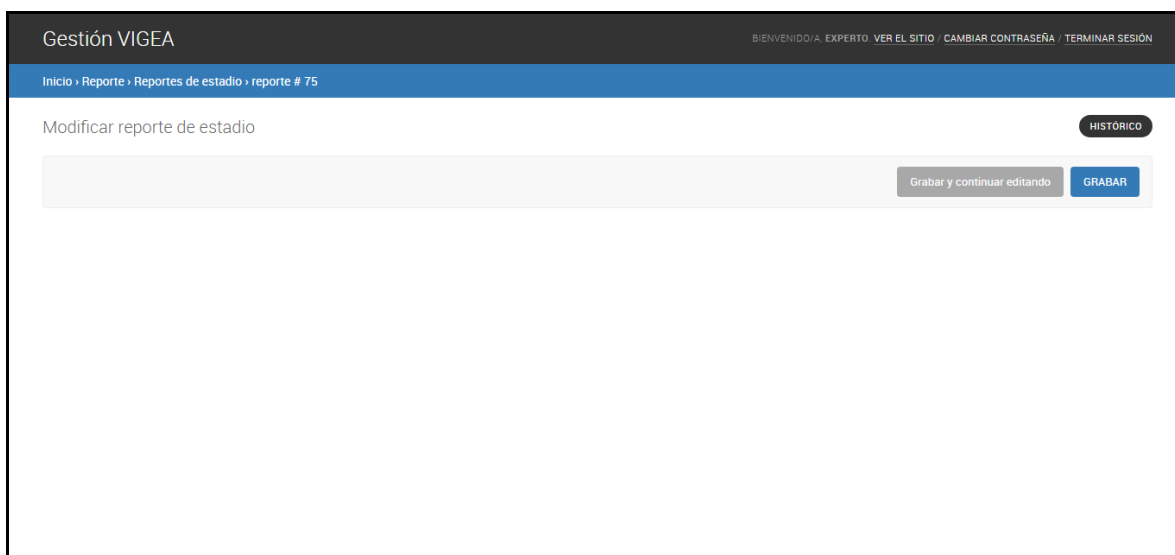


Figura 42. Error al momento en que un experto revisaba el detalle de un reporte en el sitio administrativo.

Evidenciado el error, se procedió a su respectiva corrección y como resultado se permitió a los expertos visualizar los reportes de la siguiente manera:

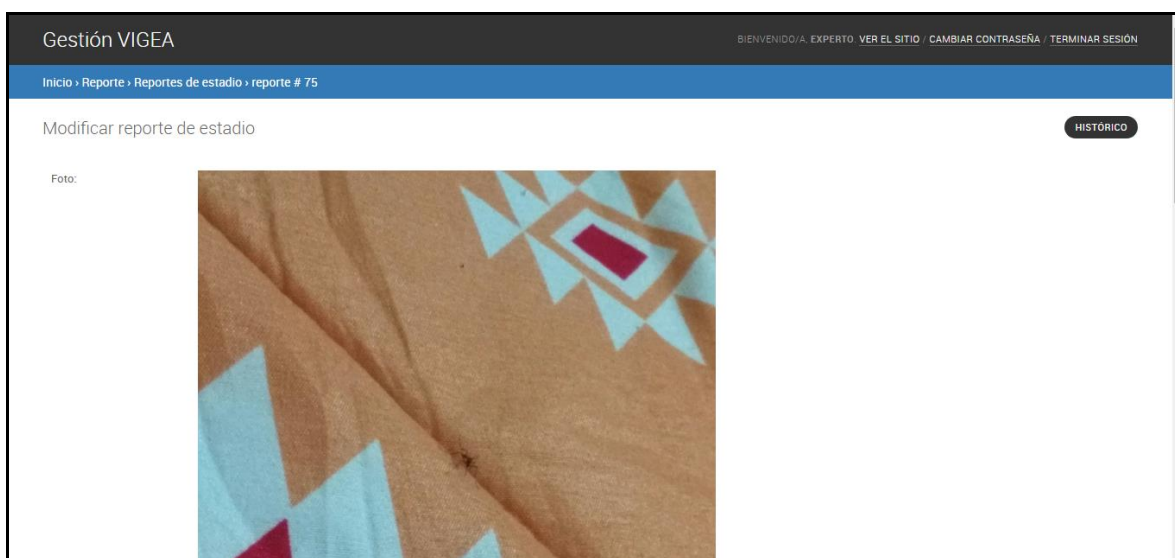


Figura 43. Detalle de un reporte en el sitio administrativo.

Coordenadas: Latitud: 4.1469594 - Longitud: -73.641144

Localización:

Fecha en la que se tomó el reporte: 24 de Mayo de 2018 a las 19:05

Descripción: N/A

Cantidad: 1

Estado: Zancudo

Este reporte es válido?

Figura 44. Detalle de un reporte en el sitio administrativo.

Durante la socialización, también se hizo una recomendación respecto al reporte de síntomas. Los participantes consideraban útil añadir un campo al reporte de síntoma en el que se reportara la edad de la persona que padecía el síntoma, pues consideraban que era relevante conocer la edad de las personas que padecían posiblemente enfermedades transmitidas por el vector *Aedes Aegypti*. De tal manera que se le añadió dicho campo al reporte y este fue el resultado:

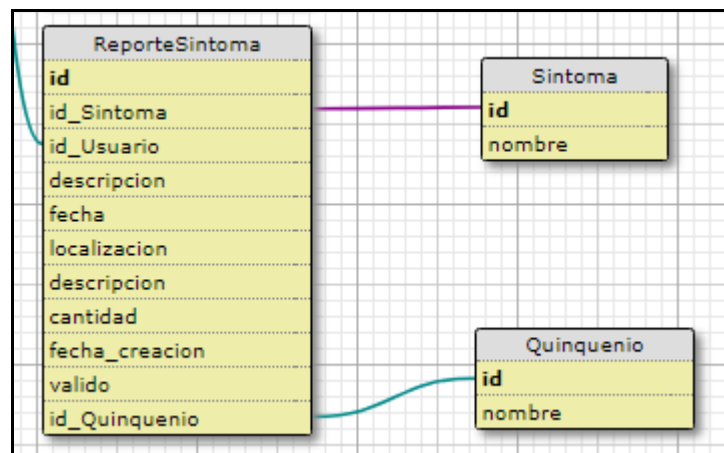


Figura 45. Representación en modelo entidad relación de la composición de las tablas, ReporteSintoma, Quinquenio, Sintoma y su relación entre ellas.

6. Realización de prueba piloto.

Corregidos los errores hallados durante la socialización del proyecto, se decidió hacer una prueba piloto, nuevamente con los estudiantes del curso de epidemiología de la Universidad de los Llanos, en esta prueba piloto, los estudiantes ejercieron nuevamente el papel de usuarios de la aplicación.

Como primera medida, la profesora del curso enseñó a los estudiantes la manera mediante la cual poder diferenciar las distintas etapas del vector *Aedes Aegypti*:



Figura 46. Identificación de las distintas etapas del vector *Aedes Aegypti*.



Figura 47. Identificación de las distintas etapas del vector *Aedes Aegypti*.



Figura 48. Identificación de las distintas etapas del vector *Aedes Aegypti*.

Luego, para los reportes recorrieron la Sede San Antonio de la Universidad de los Llanos y mediante el uso de sus teléfonos realizaron los respectivos reportes:



Figura 49. Estudiantes del programa de Enfermería creando reportes.



Figura 50. Estudiantes del programa de Enfermería creando reportes.

Como resultado, los estudiantes proveyeron los primeros reportes validos del sistema y a partir de estos datos se pudieron crear mapas de infestación.

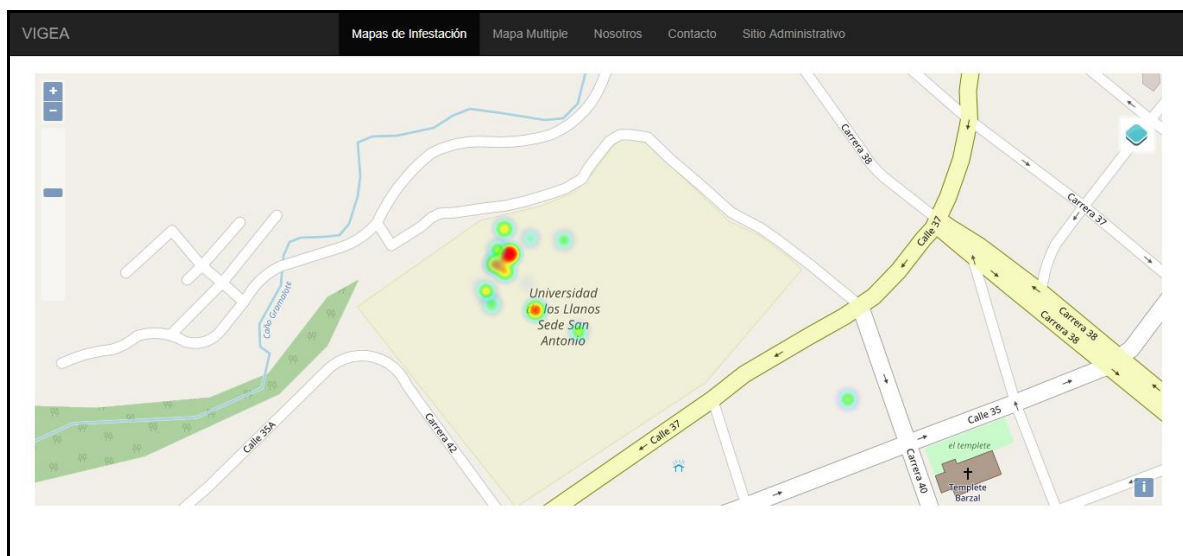


Figura 53. Mapas de infestación creados por el sistema.

RESULTADOS

- Se construyó una aplicación Web que permite observar mapas de infestación a partir de datos reportados por usuarios de una aplicación móvil, estos mapas de infestación ilustran a los usuarios lugares en los cuales se encuentran criaderos del vector *Aedes Aegypti*, ejemplares del vector o síntomas relacionados al mismo. También, la aplicación permite la gestión de dichos reportes, de sus usuarios y de los contenidos que son proveídos a los usuarios de la aplicación móvil.
- Se construyó una API que permite comunicar la aplicación Web, con una aplicación móvil, para de esta manera alimentar la base de datos y a partir de datos previamente revisados por expertos, construir mapas de infestación.
- Por medio de este proyecto se consiguió construir una aplicación Web que puede ser usada como herramienta de apoyo para la vigilancia entomológica de *Aedes Aegypti*.

CONCLUSIONES

- Con la creación de un sistema de vigilancia entomológica del vector Aedes Aegypti se permitirá a entes de control contar con una herramienta tecnológica que sirva de apoyo en la prevención del vector.
- Con la implementación de una API se logró crear un canal de información con una aplicación móvil. Mediante este canal, se espera poder alcanzar una gran cantidad de usuarios que con su participación ayudaran a crear mejores mapas de infestación.
- Mediante el desarrollo de este proyecto, se evidencio que mediante la concurrencia y tutoría de distintas áreas de conocimiento se pueden construir soluciones tecnológicas que ayuden a la comunidad.
- Este proyecto permitió fortalecer conocimientos en el lenguaje Python y comprender mejor el funcionamiento del marco de trabajo Django.
- El proceso de desarrollo de software e investigación permitió una mejor comprensión de la manera en que se desarrolla software en el mundo real.
- Realizar este proyecto permitió poner en practica lo aprendido a través de la formación universitaria en el programa de Ingeniería de Sistemas al estudiante Jorge Luis Goyeneche Mora.

TRABAJOS FUTUROS

- Análisis multitemporal de los datos capturados y almacenados por el WEBGIS.
- Técnicas de computación inteligente para validación de reportes.
- Nuevas variables en el sistema que permitan enriquecer los mapas de calor para la toma de decisiones en el sector salud.

REFERENCIAS BIBLIOGRÁFICAS.

- [1] C. A. Diaz Celis, A. A. Cruz Roa, and O. A. Gutierrez Lesmes, "Sistema de vigilancia entomológica de Aedes Aegypti basado en tecnologías mHealth, Web GIS y crowdsourcing," 2016.
- [2] M. Palaniyandi, P. Anand, and T. Pavendar, "Environmental risk factors in relation to occurrence of vector borne disease epidemics: Remote sensing and GIS for rapid assessment, picturesque, and monitoring towards sustainable health," *Int. J. Mosq. Res.*, vol. 4, no. 3, pp. 9–20, 2017.
- [3] H. A. Aziz, "A review of the role of public health informatics in healthcare," *J. Taibah Univ. Med. Sci.*, vol. 12, no. 1, pp. 78–81, Feb. 2017.
- [4] H. Chen, D. Hailey, N. Wang, and P. Yu, "A Review of Data Quality Assessment Methods for Public Health Information Systems," *Int. J. Environ. Res. Public Health*, vol. 11, no. 5, pp. 5170–5207, May 2014.
- [5] S. Luján Mora, *Programación en Internet : cliente web*. Club Universitario, 2001.
- [6] T. C. Ricketts, "Geographic Information Systems and Public Health," *Annu. Rev. Public Health*, vol. 24, no. 1, pp. 1–6, Jan. 2003.
- [7] E. K. Cromley, "GIS and Disease," *Annu. Rev. Public Health*, vol. 24, no. 1, pp. 7–24, Jan. 2003.
- [8] S. L. McLafferty, "GIS and Health Care," *Annu. Rev. Public Health*, vol. 24, no. 1, pp. 25–42, Jan. 2003.
- [9] M. Inoue, S. Hasegawa, and A. Suyama, "WEB-BASED GIS SYSTEM FOR VISUALIZING AND SHARING INFECTIOUS DISEASE INFORMATION," in *Proceedings of the IADIS International Conference on WWW/Internet*, 2009, pp. 437–439.
- [10] P. J. Ogao *et al.*, "Online GIS services for mapping and sharing disease information," *Int. J. Health Geogr.*, vol. 5, no. 1, p. 35, Feb. 2006.
- [11] M. Palaniyandi, P. Anand, and R. Maniyosai, "GIS based community survey and systematic grid sampling for dengue epidemic surveillance , control , and management : a case study of Pondicherry Municipality," *Int. J. Mosq. Res. IJMR*, vol. 1, no. 4, pp. 30–38, 2014.
- [12] K. M. N. K. B. Honda, "Web GIS in practice IV: publishing your health maps and connecting to remote WMS sources using the Open Source UMN MapServer and DM Solutions MapLab," *Int. J. Health Geogr.*, vol. 5, no. 1, p. 6, Jan. 2006.

- [13] P. Rolfhamre, K. Grabowska, and K. Ekdahl, "Implementing a public web based GIS service for feedback of surveillance data on communicable diseases in Sweden," *BMC Infect. Dis.*, vol. 4, no. 1, p. 17, Dec. 2004.
- [14] M. Palaniyandi, "Web mapping GIS: GPS under the GIS umbrella for Aedes species dengue and chikungunya vector mosquito surveillance and control," *Int. J. Mosq. Res.*, vol. 1, no. 3, pp. 18–25, 2014.
- [15] R. Jean-Baptiste *et al.*, "A Web-based GIS for health care decision-support.," *AMIA ... Annu. Symp. proceedings. AMIA Symp.*, vol. 2005, pp. 365–9, 2005.
- [16] PostgreSQL Global Development Group, "PostgreSQL: About." [Online]. Available: <https://www.postgresql.org/about/>. [Accessed: 08-May-2018].
- [17] "PostGIS — Spatial and Geographic Objects for PostgreSQL." [Online]. Available: <https://postgis.net/>. [Accessed: 10-May-2018].
- [18] GeoServer, "About." [Online]. Available: <http://geoserver.org/about/>. [Accessed: 10-May-2018].
- [19] OSGeo, "OpenLayers 3 — OSGeo-Live 11.0 Documentation." [Online]. Available: https://live.osgeo.org/es/overview/openlayers_overview.html. [Accessed: 10-May-2018].
- [20] "nginx." [Online]. Available: <https://nginx.org/en/>. [Accessed: 10-May-2018].

ANEXOS

8.1. Manual técnico

Ver imagen digital en el CD-ROM (Instrumento/manual tecnico.doc)

8.2. Manual de Usuario

Ver imagen digital en el CD-ROM (Instrumento/manual de usuario.doc)

8.3. Casos de Uso

Ver imagen digital en el CD-ROM (Instrumento/casos de uso.uml)

8.4. Modelo Entidad-Relación construido en MySQL Workbrench

Ver imagen digital en el CD-ROM (Instrumento/mer en workbrench.mwb)

FACULTAD DE CIENCIAS BASICAS E INGENIERIA

AUTORIZACIÓN

Yo **Jorge Luis Goyeneche Mora** mayor de edad, identificado con la Cédula de Ciudadanía No. **1.121.925.828** de **Villavicencio**, actuando en nombre propio en mi calidad de autor del trabajo de grado denominado **Sistema de información geográfica en la web (WEBGIS) para el apoyo a la vigilancia entomológica de Aedes Aegypti**, hago entrega del ejemplar y de sus anexos de ser el caso, en formato digital o electrónico (CD-ROM) y autorizo a la **UNIVERSIDAD DE LOS LLANOS**, para que en los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia, con la finalidad de que se utilice y use en todas sus formas, realice la reproducción, comunicación pública, edición y distribución, en formato impreso y digital, o formato conocido o por conocer de manera total y parcial de mi trabajo de grado o tesis.

Yo **JORGE LUIS GOYENECHÉ MORA**, Como autor, manifiesto que el trabajo de grado o tesis objeto de la presente autorización, es original y se realizó sin violar o usurpar derechos de autor de terceros; por tanto, la obra es de mi exclusiva autoría y poseo la titularidad sobre la misma; en caso de presentarse cualquier reclamación o acción por parte de un tercero en cuanto a los derechos de autor sobre la obra en cuestión, como autor, asumiré toda la responsabilidad, y saldré en defensa de los derechos aquí autorizados, para todos los efectos la Universidad actúa como un tercero de buena fe.

Para constancia, se firma el presente documento en dos (2) ejemplares del mismo valor y tenor en Villavicencio - Meta, a los 3 días del mes de agosto de dos mil dieciocho (2018).

JORGE LUIS GOYENECHÉ MORA

Firma _____

Nombre: _____

C.C No. _____ de _____